REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
12-12-2014	MS Thesis	-

4. TITLE AND SUBTITLE
Measuring the Influence of Mainstream Media on Twitter Users

5a. CONTRACT NUMBER
W911NF-11-1-0168

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
633002

6. AUTHORS
Omar Eltayeby

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
North Carolina A&T State University
1601 East Market Street

Greensboro, NC 27411 -0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
60143-ST-H.22

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
This thesis is based on a research project that has been conducted at Clark Atlanta University (CAU) under the supervision of Professors Roy George and Peter Molnar. The aim of the project is to construct a framework for measuring the influence of mass media on Twitter users. Media influence or media effects are used in media studies, psychology, communication theory and sociology to refer to the theories about the ways in which mass media and media culture affect how their audiences think and behave. Arguably, the

15. SUBJECT TERMS
mainstream media, data analysis, twitter

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Ajit Kelkar
UU	UU	UU	UU		19b. TELEPHONE NUMBER 336-285-2864

Report Title

Measuring the Influence of Mainstream Media on Twitter Users

ABSTRACT

This thesis is based on a research project that has been conducted at Clark Atlanta University (CAU) under the supervision of Professors Roy George and Peter Molnar. The aim of the project is to construct a framework for measuring the influence of mass media on Twitter users. Media influence or media effects are used in media studies, psychology, communication theory and sociology to refer to the theories about the ways in which mass media and media culture affect how their audiences think and behave. Arguably, the agenda-setting process is an unavoidable part of news gathering by the large

MEASURING THE INFLUENCE OF MAINSTREAM MEDIA ON TWITTER USERS

A DISSERTATION

SUBMITTED TO THE FACULTY OF CLARK ATLANTA UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE MASTERS DEGREE

BY OMAR ELTAYEBY

DEPARTMENT OF COMPUTER & INFORMATION SCIENCES

ATLANTA, GEORGIA

JULY 2014

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

This thesis is based on a research project that has been conducted at Clark Atlanta University (CAU) under the supervision of Professors Roy George and Peter Molnar. The aim of the project is to construct a framework for measuring the influence of mass media on Twitter users. Media influence or media effects are used in media studies, psychology, communication theory and sociology to refer to the theories about the ways in which mass media and media culture affect how their audiences think and behave. Arguably, the agenda-setting process is an unavoidable part of news gathering by the large organizations which make up much of the mass media. For example, four main news agencies — AP, UPI, Reuters and Agence-France-Presse — together provide 90% of the total news output of the world's press, radio and television[1]. According to Stuart Hall, because some of the media produce material which often is impartial and serious, they are accorded a high degree of respect and authority. Stuart says, "independence is not a mere cover, it is central to the way power and ideology are mediated in societies like ours" (Stuart Hall, 1973). In 1972, McCombs and Shaw demonstrate the agenda-setting effect at work in a study conducted in Chapel Hill, North Carolina, USA during the 1968 presidential elections. A representative sample of un-decided voters was asked to outline the key issues of the election as it perceived them. Concurrently, the mass media serving

[1] http://newint.org/features/1981/06/01/four/

these subjects were collected and their content was analyzed. The results showed a definite correlation between the two accounts of predominant issues. The purpose, of this current study on the application level, shows the same correlation, but between the mass media and the people's opinion through twitter.

On the development level, the basic concept of finding this correlation derives the methodology for our analyzing the sentiment used on Twitter. A comparison between the sentiment used when mentioning and not mentioning news sources on Twitter towards trending topics is shown to infer the how much the mass media is influential. In Computer Science, sentiment analysis (also known as opinion mining) refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. A basic task in sentiment analysis (Michelle de Haaf, 2010) is classifying the polarity of a given text at the document, sentence, or feature/aspect level, whether the expressed opinion in a document, a sentence or an entity feature/aspect is positive, negative, or neutral. Advanced, "beyond polarity" sentiment classification looks, for instance, at emotional states such as "angry," "sad," and "happy" (Linhao Zhang, 2013). Many research works were done in the field of aspect-based opinion mining on scientific documents, web content generally and social media for multiple purposes such as stock market sentiment analysis, opinion mining about product

features, spam review detection etc…. The aspect-based opinion mining task in this project is accomplished first by extracting the topics which is mostly concerned by the twitter users then finding the semantic relatedness between the corresponding words used to describe those topics. Concretely, semantic relatedness can be estimated for instance by defining a topological similarity, by using ontologies to define a distance between terms. The ontology of terms could be defined by several text corpuses, which we used in our project by importing them using the Natural Language Processing Toolkit (NLTK). As an example, a naive metric for the comparison of concepts ordered in a partially ordered set and represented as nodes of a directed acyclic graph (taxonomy), would be the minimal distance in terms of edges composing the shortest-path linking the two concept nodes. Based on text analyses, semantic distance between units of language can also be estimated using statistical means such as a vector space model to correlate words and textual contexts from a suitable text corpus (co-occurrence).

The remainder of this thesis is organized in the following manner. Chapter 2 is the literature review to show previous related work from other papers and projects in the field of text mining, association rules mining, sentiment analysis and opinion clustering. This chapter will not handle the steps or the work done in the project, it will just cover a broad perspective of different applications and work done in those areas. Such exposure to other work enhances the readers awareness about the contribution of this thesis to the various fields mentioned. Chapter 3 shows the framework in details and the previous analysis

done before constructing the framework. This analysis discusses some of the primary results that are the outcome of the initial framework. This chapter includes the methodology of collecting the data from twitter on the fedora cluster of CAU, some statistical analysis that shows the general trend of tweets' types that users post and the framework of the aspect-based opinion mining process. The chapter handles the details of the steps specified in the framework with an explanation of how we fit the algorithms used into our model. Through chapter 4 inter-step results are shown with the visuals that show the meaning and inferences about the results by discussing those visuals and how they fulfill the aim of this project on both the application and computer science levels. Finally, chapter 5 concludes the research thesis through explaining the best practices for developing this framework and the disadvantages that were encountered out of these results.

CHAPTER 2

LITERATURE REVIEW

Introduction to Text Mining:

In recent times, the amount of textual information available in electronic form is growing at staggering rate. The best example of this growth is the World Wide Web (WWW), which is estimated to provide access to Exabytes of text. Even in commercial and private hands text collection sizes which were unimaginable a few year ago are common now, and the challenge is to efficiently mine interesting patterns, trends and potential information that are of interest to the user (Ricardo Baeza-Yates et al., 2002). Text mining, also known as Intelligent Text Analysis, Text Data Mining or Knowledge-Discovery in Text (KDT), refers generally to the process of extracting interesting and non-trivial information and knowledge from unstructured text. In general, data mining deals with structured data (for example relational databases), whereas text presents special characteristics and is unstructured. The unstructured data is totally different from databases, where mining techniques are usually applied and structured data is managed (Vishal Gupta and Gurpreet S. Lehal, 2009). Text mining could be used for unstructured or semi-structured data sets such as emails, full-text documents and HTML files and more (Delgado et al., 2002). An example of semi-structured data is an email with appointment details, holding information about the location, time and date. This type of information formats are easier to analyze for whatever purposes due to the organized and

rigid data types used in such cases. Text mining shares many characteristics with classical data mining, but differs in many ways (Ah-hwee Tan, 1999).

- Many knowledge discovery algorithms defined in the context of data mining, are irrelevant or ill-suited for the textual application
- Special mining tasks, such as concept relationship analysis, are unique to text mining
- The unstructured form of the full text necessitates special linguistic pre-processing for extracting the main features of the text

Text mining is a multidisciplinary field, involving information retrieval, text analysis, information extraction, clustering, categorization, visualization, database technology, machine learning, and data mining (Nasukawa and Nagano, 2001). In text mining approaches, initially the unstructured text documents are processed using Natural Language Processing (NLP) techniques to extract keywords labeling the items in that text documents. Then, classical data mining techniques are applied on the extracted data (keywords) to discover interesting patterns. Starting with a collection of documents, a text mining process would retrieve a particular document and preprocess it by checking format and character sets. Then it would go through a text analysis phase, sometimes repeating techniques until information is extracted (Vishal Gupta and Gurpreet S. Lehal, 2009).

Text preprocessing classically means tokenization and then Part of Speech Tagging, as we will see in the methodology chapter how we used the NTLK, or in a bag of words approach word stemming and the application of a stop word list. Tokenization is the process of splitting the text into words or terms. Part of Speech (PoS) Tagging tags words according to the grammatical context of the word in the sentence, hence dividing up the words into nouns, verbs and more. This is important for the exact analysis of relations between words, as it is needed in the extraction of relations between the texts. Most text mining objectives fall under the following categories of operations: Search and Retrieval, categorization (supervised classification), summarization, Trends Analysis, Associations Analysis, Visualization and more.

The following subsections explain different techniques for information extraction from text documents generally, while focusing on previous work related to our research project in the area of Association Rule Mining, Temporal Association Rules Mining, Prototypical Documents Mining, Clustering and other text mining.

Information Extraction:

Related work in Association Rule mining:

Association is a powerful data analysis technique that appears frequently in data mining literature (Pack Chung et al., 1999). Today the mining of such rules is still one of the most popular pattern-discovery methods in Knowledge Discovery & Data Mining

(KDD). Association Rule Mining is the process of discovering collection of data attributes that are statistically associated in the underlying data. Association rules aim to extract interesting correlations, frequent patterns, associations or causal structures among sets of items in the transaction databases or other repositories. First, minimum support is applied to find all frequent itemsets in a database. In second step, the frequent item sets and the minimum confidence constraint are used to form rules. The main advantages of association rules are simplicity, intuitiveness and freedom from model-based assumptions. The important application of association rule mining is market basket analysis which is a famous tool among retail enterprises, for example they inform the user about items most likely to be purchased by a customer during a visit to the retail store. They are widely used in many other areas such as telecommunication networks, market and risk management, inventory control and more (Qiankun Zhao et al., 2013).

Information extraction systems can be used to directly extricate abstract knowledge from a text corpus, or to extract concrete data from a set of documents which can then be further analyzed with traditional data mining techniques to discover more general patterns (Raymond Mooney and Razvan Bunescu, 2005). In short, information extraction is the task of locating desired pieces of data from a document. Many text mining methods have been developed in order to achieve the goal of retrieving useful information for users, for example the paper published by Edda and Jorg and Sebastiani in 2002. Most text mining methods use the keyword- based approaches, whereas others choose the phrase technique to construct a text representation for a set of documents. It is

believed that the phrase-based approaches should perform better than the keyword- based ones as it is considered that more information is carried by a phrase than by a single term. Based on this hypothesis, Lewis in 1992, conducted several experiments using phrasal indexing language on a text categorization task. The results showed that the phrase-based indexing language was not superior to the word-based one. Although phrases carry less ambiguous and more succinct meanings than individual words, the likely reasons for the discouraging performance from the use of phrases are:

1. Phrases have inferior statistical properties to words
2. They have a low frequency of occurrence
3. There are a larger number of redundant and noisy phrases among them

In recent times, extracting semantic relationships among entities in text documents has gained enormous popularity. However, its application on text databases is still very challenging because characteristics of text and transaction databases are different. This leads to the motivation of our research, that is to apply association rule mining to text databases to capture the relationships among words (terms). Association rules have been researched and applied extensively, in diverse domains and applications (Bench-Capon et al., 2000). In text mining, extracted rules can be interpreted as co-occurrences of terms in texts and consequently are able to reflect semantic relations between terms. However, it should be mentioned that the association rule extraction is of exponential growth and a very large number of rules can be produced. The extracted association rules identify the

relations between features in the documents collection. The scattering of features in text contribute to the complexity of define features to be extracted from text. These kinds of features relationships can be better described with the association rule mining of text. Several researchers have presented algorithms and approaches for mining associations from text document collections, for example Hany Mahgoub in 2006.

A handful of text mining approaches are available in the literature for mining potential information and associations from large collections of text documents. A brief review of some recent researches related to mining associations from text documents is presented here:

Hany Mahgoub et al. in 2006 have described that the text mining technique for automatically extracting association rules from collections of textual documents. The technique called, Extracting Association Rules from Text (EART). It depends on keyword features for discover association rules amongst keywords labeling the documents. In their work, the EART system ignores the order in which the words occur, but instead focusing on the words and their statistical distributions in documents. The main contributions of the technique are that it integrates XML technology with Information Retrieval scheme Term Frequency-Inverse Document Frequency (TF-IDF) (for keyword/feature selection that automatically selects the most discriminative keywords for use in association rules generation) and use Data Mining technique for association rules discovery. Experiments applied on Web Pages news documents related

to the outbreak of the bird flu disease. The extracted association rules contain important features and describe the informative news included in the documents collection. The performance of the EART system compared with another system that has been used the Apriori algorithm throughout the execution time and evaluating extracted association rules.

Liang-Chih Yu et al. in 2011 have proposed a framework that combines a supervised data mining algorithm and an unsupervised distributional semantic model to discover association language patterns. The association rule mining was used to generate a set of seed patterns by incrementally associating frequently co-occurring words from a small corpus of sentences labeled with negative life events. The distributional semantic model was then used to discover more patterns similar to the seed patterns from a large, unlabeled web corpus.

Suneetha Manne, and Sameen Fatima in 2011 have proposed the method of Text Categorization on web documents using text mining and information extraction based on the classical summarization techniques. First web documents were preprocessed to establish an organized data file, by recognizing feature terms like term frequency count and weight percentage of each term. Experimental results were showed, that approach of Text Categorization was more suitable for informal English language based web content where there was vast amount of data built in informal terms. That method had

significantly reduced the query response time, improved the accuracy and degrees of relevancy.

Pablo F. Matos et al. in 2010 have addressed the problem of extracting and processing relevant information from unstructured electronic documents of the biomedical domain. The documents were full scientific papers. That problem imposed several challenges, such as identifying text passages that contain relevant information, collecting the relevant information pieces, populating a database and a data warehouse, and mining these data. For that purpose, that paper has proposed the IEDSS-Bio, an environment for Information Extraction and Decision Support System in Biomedical domain. In a case study, experiments with machine learning for identifying relevant text passages (disease and treatment effects, and patients number information on Sickle Cell Anemia papers) showed that the best results (95.9% accuracy) were obtained with a statistical method and the use of preprocessing techniques to resample the examples and to eliminate noise.

Chenn-Jung et al. in 2010 have proposed a financial news headline agent to assisting the investors in deciding to buy and to sell stocks in Taiwan market after receiving the essential real-time news headline disseminated by the agent. Weighted association rules and text mining techniques were used to derive the significance degree of each newly arrived news headline on the fluctuation of Taiwan Stock Exchange Financial Price Index on the next trading day. The experimental results revealed that the

proposed work indeed achieves significant performance and demonstrate its feasibility in the applications of real-time information dissemination, such as financial news headlines via Internet.

Sophia Ananiadou Jung et al. in 2010 have summarized the methods that were currently available, with a specific focus on protein–protein interactions and pathway or network reconstruction. The approaches described will be of considerable value in associating particular pathways and their components with higher-order physiological properties, including disease states.

Yue Dai et al. 2011 have proposed MinEDec, a decision-support model that combines two well- known and widely-used Competitive Intelligence (CI) analysis models into a unified model. CI analysis by using this unified model was supported by the use of state-of-the-art text mining technologies. They have also outlined the architecture of a Decision Support System that was based on the MinEDec model and applies various text mining technologies. First, they explained that the purpose of the MinEDec model was to transform data into useful knowledge. They then described the functions of Strengths, Weaknesses, Opportunities and Threats (SWOT) analysis and the Five Force Analysis framework in a new model for monitoring the business environment. Although there were several CI software tools available, none of them combines text mining and several widely accepted CI analysis methods. The proposed model was unique as it analyses the five objectives from the perspective of nine SWOT factors by

using text mining technologies. Based on this, they have proposed a way of integrating SWOT and FFA models into a unified decision- support model.

Fei Wu and Daniel Weld in 2010 presented Wikipedia-based Open Extractor (WOE), an open information extraction system which improves dramatically on TextRunner's precision and recall. The key to WOE's performance was a form of self-supervised learning for open extractors - using heuristic matches between Wikipedia infobox attributes values and corresponding sentences to construct training data. Like TextRunner, WOE's extractor eschews lexicalized features and handles an unbounded set of semantic relations. WOE was operated in two modes: when restricted to part-of-speech (POS) tag features, it runs as quickly as TextRunner, but when set to use dependency-parse features its precision and recall rise even higher.

Prototypical Document Extraction:

Another direction of research for automated information extraction is to apply knowledge discovery techniques to the complete textual content of the documents (in a so called "full text" approach as opposed to approaches only considering indexing key-words). However, experiments on the Reuter corpus (Rajman, 1997) have shown that the extraction process does not produce any exploitable results when the standard association extraction techniques are directly applied on the words contained in the documents instead of operating on the already abstract concepts represented by the key words

(Mosley and Roosvelt, 2012). Among the extracted associations, some only indicate the presence of domain dependent compounds.

Therefore, a different approach is necessary when full text is considered; Prototypical Document Extraction. A prototypical document is informally defined as a document corresponding to information that occurs in a repetitive fashion in the document collection, i.e. a document representing a class of similar documents in the textual base. The extraction techniques operating in such framework still use the notion of frequent sets, but additional NLP techniques are used to preprocess the data, and identify more significant linguistic entities (terms) for frequent set extraction process. More, precisely, the NL preprocessing, realized in collaboration with R. Feldman's team at Bar Ilan University, was decomposed into two steps: Part-of-Speech tagging and Term Extraction.

Part-of-Speech tagging:

This process automatically identifies the Morpho-syntactic categories (noun, verb, adjective etc...) of words in the documents. Such tagging allows filtering non-significant words on the basis of their morpho-syntactic category. In our experiments, we used a rule-based tagger designed by E.Brill (Brill, 1992) that is implemented in the Natural Language Processing Toolkit functions, and restricted the extraction process to operate only on nouns, verbs, adjectives and adverbs.

Term extraction:

This process aims at the identification of the domain-dependent compounds. It allows the mining process to focus on more meaningful co-occurrences, and can be decomposed into: term candidates' identification (on the basis of structural linguistic term candidates filtering (based on statistical relevance scoring (Daille, 1994)).

Clustering & other techniques:

Several other domains concerned with Textual Data Processing (such as Textual Data Analysis or Content Analysis) can provide interesting insights on the techniques presented in this literature review. The problem of frequent set extraction could be for instance partially related to the identification of co-occurring words (Lafon,1981), repeated segments (Salem,1987), or quasi-segments (Becue,1993), often considered in the domain of Textual Data Analysis. The main difference here is that the Text Mining techniques rely on the use of frequencies of sets of words instead of considering co-frequencies of pairs. As far as more sophisticated information extraction is concerned, methods used in Textual Data Analysis (Lebart, 1998) usually rely on a cluster analysis based on the chi-square distance between the lexical profiles. For each of the resulting clusters of documents, characteristic words (Lafon, 1980) (i.e. words with a frequency in the cluster significantly higher than the one expected according to a predefined probabilistic model) are then extracted. Each of the clusters is then represented by a characteristic document which is the document in the cluster that contains the most

characteristic words. The differences between such approaches and prototypical document extraction as described in this section are essentially of two kinds:

1. Prototypical document extraction integrates a more substantial amount of explicit linguistic knowledge, in particular in the preprocessing phase, where morpho-syntactic patterns are used for the extraction of indexing terms
2. The aims underlying the two methods are in fact quite different: documents characteristic for a cluster identify the information content that is the more discriminant for the cluster relatively to the rest of the document collection.

On the opposite, prototypical documents tend to identify repetitive patterns of texts particularly frequent in the document collection, and that will serve to structure its informational content. The two approaches therefore appear to be rather complementary in the sense that prototypical documents could be thought as kinds of linguistic frames in which the informational content (as identified by the characteristic documents) could be preferentially expressed.

In addition, in order to allow better representatives, a more generic representation could be achieved by using name entity tagging, a semantic tagging that allows to identify and generalize certain elements of a sentence. Such a tagging could lead to representations where the variable parts of the prototypical documents would be replaced by concepts.

Social Media Mining:

Social Media Background:

Regardless of where you look, you can see an exaggeration in the use of social media. Online communities have developed that focus on both personal and professional lives. Groups have been formed that focus on every potential area of interest, including food, sports, music, parenting, scrapbooking, and actuarial issues. It is estimated that there are over 900 social media sites on the internet. Some of the more popular platforms are Facebook, Twitter, LinkedIn, Google Plus, and YouTube. To help understand the explosion in the use of social media, consider the following statistics which were compiled in November 2013 by Jonathan Bernstein[1].

- There are 751 million users on Facebook from mobile with 7,000 different devices.
- There are over 288 million monthly active users on Twitter.
- Over 343 million active users on Google+.
- Total number of LinkedIn groups is 1.5 million.
- Seventy-seven percent of internet users read blogs.

The majority of the population is using social media in some form or another. Given the substantial increase in the use of social media, there is a significant amount of

[1] http://socialmediatoday.com

information that is being generated. As seen in the same sources referenced above, the volume of this content is staggering:

- 350 million Photos are uploaded every day.
- There are over 1 billion unique monthly visitors on YouTube.
- On an average, over 400 million tweets are being sent per day.
- Over 3 million LinkedIn company pages.

So not only are people joining and accessing social media sites, but they are also spending time engaging in social media and creating a significant amount of content. As a result of this time spent on social media and the information being generated, businesses have taken notice and are attempting to leverage the power of social media to help them succeed[2].

- Two-thirds of comScore's U.S. Top 100 websites and half of comScore's Global Top 100 websites have integrated with Facebook.
- Many businesses now have established Twitter accounts in an attempt to connect with current and potential customers.
- Eighty-eight percent of companies use LinkedIn as a recruitment tool.
- Corporate blogging accounts for 14% of blogs.

[2] wealthinvest.com

The commitment that businesses are making to increase their presence in social media is also being shown in the resources they are committing to this effort. According to eMarketer, U.S. advertisers increased the digital ad spending. As digital matures, and continues siphoning dollars from traditional media, the options within digital advertising are also proliferating. Breaking down where advertisers expected to make the biggest web increases, social media advertising ranked first, with 47% of respondents expecting to up investments in the next year[3]. According to Banking2020.com, 50% of Chief Marketing Officers at Fortune 1000 companies say they have launched a corporate blog because it is a cost of doing business today. So not only is the corporate investment being evidenced by dollars spent but also in the time it takes to create and maintain social media efforts.

Twitter Background:

Twitter is a social networking site that allows users to send and read short messages of a maximum of 140 characters. Twitter was created in March 2006 and was officially launched in July 2006. The growth of Twitter has been phenomenal, as was shown by the facts mentioned in the previous section. Users sign up for an account on Twitter, and once they have an account they can begin to "tweet," which is the

[3] http://www.emarketer.com/Article/Social-Video-Sites-Will-See-Big-Boosts-US-Advertiser-Spending/1010300#z5CFEMvLICRf2uvU.99

terminology for sending a message. Users can subscribe to other user's tweets, a process known as "following." These subscribers are known as "followers." By default, tweets that a user sends are public to everyone; however, users can also choose to send tweets specifically to their followers that will not be visible to the public.

Users on Twitter are identified by a user name, and this user name is proceeded by the "@" symbol. When a user identifies another user in their tweet by their user name, it will be visible to the public, and the user that is referenced will be notified by Twitter that they have been "mentioned." If a user sees a tweet that is interesting and wants to pass the information along, they can "retweet" the post, which is similar to forwarding an email message to a new set of users, in this case their followers. Retweets will generally be identified with an "RT" that is embedded in the message. Messages can be grouped by topic or type by the use of hashtags "#". A hashtag preceding the topic will allow Twitter users to find tweets related to a particular topic when performing a search. Twitter also has a location function. If users are tweeting from a mobile device, they can choose to turn on their location, and their latitude and longitude will be captured with the tweet.

Tweets can be related to anything, but much of the content on Twitter is related to several key categories. These categories were outlined in research done by Pear Analytics[4]. This study found that tweets were primarily related to six categories:

- Pointless babble

[4] http://www.pearanalytics.com/blog/

- Conversational
- Pass along value
- Self-promotion
- Spam
- News

Twitter is a conduit for many different types of information, including breaking news (Kwak et al. 2010), political discourse (Conover et al. 2010), community events (Washington Post 2011a), and call for protest (Los Angeles Times 2011). Twitter's reach and diversity of uses makes it a powerful tool for shaping public opinion: indeed Twitter is already being used to defame political candidates and discredit their views (Ratkiewicz et al. 2010; Metaxas and Mustafaraj 2010). Countries such as China are using censors to track internet discussions and shape opinions. Brigham Young University[5], most people who closely follow both political blogs and traditional news media tend to believe that the content in the blogosphere is more trustworthy.

There have been many research applications and challenges proposed in the knowledge discovery conferences for facilitating social media, Twitter particularly, to mine, detect, identify, cluster and classify useful information about Twitter users. Such information could be used by marketing companies, news agencies, governments etc …

[5] http://news.byu.edu/archive09-may-blogs.aspx

for different interests and uses. The following is a summary of papers in the field of mining social media data to exploit the general direction of such field:

Roosevelt C. Mosley Jr in 2012 discussed various applications of correlation, clustering and association analyses to social media for insurance companies. The paper demonstrates the analysis of insurance Twitter posts to help identify keywords and concepts which can facilitate the application of this information by insurers. As insurers analyze this information and apply the results of the analysis in relevant areas, they will be able to proactively address potential market and customer issues more effectively. The paper also proposes the challenges faced in the process of analyzing social media data such as accessing, collecting and cleaning the data, which is a big dilemma in most social media projects.

Xintian Yang et al. in 2012 presented a dynamic pattern driven approach to summarize data produced by Twitter feeds. The developed novel approach maintains an in-memory summary while retaining sufficient information to facilitate a range of user-specific and topic-specific temporal analytics. Also, in this paper they compare their approach with several state-of-the-art pattern summarization approaches along the axes of storage cost, query accuracy, query flexibility, and efficiency using real data from Twitter. Their approach is found not only scalable but also outperforms existing approaches by a large margin.

Hila Becker et al. in 2011 explored approaches for analyzing the stream of Twitter messages to distinguish between messages about real-world and non-event messages. The approach relies on a rich family of aggregate statistics of topically similar message clusters based on temporal, social, topical and Twitter-centric features. The authors use these features to develop query formulation strategies for retrieving content associated with an event on different social media sites. Further, they explore ways in which event content identified on one social media site can be used to retrieve additional relevant event content on other social media sites. They apply the strategies to a large set of user-contributed events, and analyze their effectiveness in retrieving relevant event content from Twitter, YouTube, and Flickr. The results of large-scale experiments over millions of tweets the effectiveness in the approach for surfacing real-world event content on Twitter.

Geli Fei et al. in 2013 approached the problem of automatic spam detection of reviews by exploiting the burstiness of nature of reviews to identify the review spammers. The reviewers and their occurrence in bursts are modeled as a Markov Random Field (MRF), and employ the Loopy Belief Propagation (LBP) method to infer whether a reviewer is a spammer or not in the graph. The paper proposes several features and employ feature induced message passing in the LBP framework for network inference. Additionally, the paper proposes a novel evaluation method to evaluate the detected spammers automatically using supervised classification of their reviews. The authors employ domain experts to perform a human evaluation of the identified

spammers and non-spammers. Both the classification result and human evaluation result show that the proposed method outperforms strong baselines, which demonstrate the effectiveness of the method.

Zhiyuan Chen et al. in 2013 proposed the problem of identifying intention posts in online discussion forums. The author exploits several special characteristics of the problem using a new transfer learning method unlike the general ones used in other research problems. The paper starts with discussing the Expectation Maximization algorithm and its Feature Selection version, and finally the Co-Class algorithm which is inspired by Co-training in (Blum & Mitchell, 1998).

Arjun Mukherji and Bing Liu 2012 proposed the problem fine-grained mining of contentions in discussion/debate forums. The goal of this paper is to discover contention and agreement indicator expressions, and contention points or topics both at the discussion collection level and also at each individual post level. The paper proposes three models to solve the problem, which not only model both contention/agreement expressions and discussion topics, but also, more importantly, model the intrinsic nature of discussions/debates, i.e., interactions among discussants or debaters and topic sharing among posts through quoting and replying relations. Evaluation results using real-life discussion/debate posts from several domains demonstrate the effectiveness of the proposed models.

Trending Topics:

In the previous related work discussed we exposed the characteristics and the research done on Twitter and social media generally, this subsection discusses the special work done on discovering the trending topics. Our research will later address the same topic of trending topics on Twitter but using a different technique.

Glivia Barbosa et al. in 2012 described the preliminary results and future directions of a research in progress, which aims at assessing the hashtag effectiveness as a resource for sentiment analysis expressed on Twitter. The results so far support our hypothesis that hashtags may facilitate the detection and automatic tracking of online population sentiment about different events. This hypothesis shapes our research as will be shown in the methodology chapter towards using hashtags the basic input for finding trending topics on Twitter.

Yiye Ruan et al. in 2012 discussed an approach for predicting microscopic (individual) and macroscopic (collective) user behavioral patterns with respect to specific trending topics on Twitter. The paper seeks to predict the strength of content generation which allows more accurate understanding of Twitter users' behavior and more effective utilization of the online social network for diffusing information. While previous efforts have been focused on analyzing driving factors in whether and when a user will publish topic-relevant tweets. The paper considers multiple dimensions into one regression-based prediction framework covering network structure, user interaction, content characteristics

and past activity. Experimental results on three large Twitter datasets demonstrate the efficacy of the proposed method. They find in particular that combining features from multiple aspects (especially past activity information and network features) yields the best performance. Furthermore, they observe that leveraging more past information leads to better prediction performance, although the marginal benefit is diminishing.

Chi Wang et al. 2013 presented an algorithm for recursively constructing multi-typed topical hierarchies for constructing high quality concept hierarchies that can represent topics at multiple granularities benefits tasks such as search, information browsing, and pattern mining. The idea is based on modelling heterogeneous digital data collections as a heterogeneous information network, linking text with multiple types of entities. The proposed approach handles textual phrases and multiple types of entities by a newly designed clustering and ranking algorithm for heterogeneous network data, as well as mining and ranking topical patterns of different types. Their experiments on datasets from two different domains demonstrate that the algorithm yields high quality, multi-typed topical hierarchies.

Mor Naaman et al. in 2011 made two essential contributions for interesting interpreting emerging temporal trends in these information systems First, based on a large dataset of Twitter messages from one geographic area, they developed a taxonomy of the trends present in the data. Second, they identified important dimensions according to which trends can be categorized, as well as the key distinguishing features of trends that

can be derived from their associated messages. They quantitatively examine the computed features for different categories of trends, and establish that significant differences can be detected across categories. Their study advances the understanding of trends on Twitter and other social awareness streams, which will enable powerful applications and activities, including user driven real-time information services for local communities.

Sentiment Analysis:

Zhiyuan Chen et al. in 2013 proposed a framework to leverage the general knowledge in topic models. Such knowledge is domain independent. Specifically, they use one form of general knowledge, i.e., lexical semantic relations of words such as synonyms, antonyms and adjective attributes, to help produce more coherent topics. However, there is a major obstacle, i.e., a word can have multiple meanings/senses and each meaning often has a different set of synonyms and antonyms. Not every meaning is suitable or correct for a domain. Wrong knowledge can result in poor quality topics. To deal with wrong knowledge, they proposed a new model, called GK- LDA, which is able to effectively exploit the knowledge of lexical relations in dictionaries. There experiments using online product reviews show that GK- LDA performs significantly better than existing state-of-the-art models. We expose such research since we are going

to show how we used lexical semantic relations from synonym lists for sentiment analysis, which is a bottleneck in our project.

Carmela Cappelli in 2003 focused on synonym relations between words. A cluster analysis approach is presented, aiming at detecting groups of synonyms of a given term which are characterized by a high degree of homogeneity and therefore are interchangeable. Some applications to the case of Italian words are shown and discussed. The results show that the proposed approach is promising in identifying different senses of a word. In relation to our work this paper exposes the use of hierarchical clustering for appealing the Dendogram of relations between words driven by synonym list.

Seungyeon Kim et al. in 2012 considered higher dimensional extension of the sentiment concept which represent a richer set of human emotions. The approach's model contains a continuous manifold rather than a finite set of human emotions. The paper investigated the resulting model, compared it to psychological observations, and explored its predictive capabilities. Besides obtaining significant improvement over a baseline without manifold, the paper showed a visualization of different notions of positive sentiment in different domains.

Elif Aktolga et al. in 2013 focused on diversifying the sentiment according to explicit bias to allow users to switch the result perspective to better grasp the polarity of opinionated content, such as during a literature review. For this, the paper first inferred the prior sentiment bias inherent in a controversial topic - the `Topic Sentiment'. Then,

utilized this information in 3 different ways to diversify results according to various sentiment biases: Equal diversification to achieve a balanced and unbiased representation of all sentiments on the topic; Diversification towards the Topic Sentiment, in which the actual sentiment bias in the topic is mirrored to emphasize the general perception of the topic; Diversification against the Topic Sentiment, in which documents about the 'minority' or outlying sentiment(s) are boosted and those with the popular sentiment are demoted. In the same sense our research direction, towards sentiment value assignment stage, changed to use scoring and lexical semantic relations instead of positive and negative word lists.

Johan Bollen et al. in 2011 investigated the correlation between the collective mood states derived from large-scale Twitter feeds and the value of the Dow Jones Industrial Average (DJIA) over time. They analyzed the text content of daily Twitter feeds by two mood tracking tools, namely OpinionFinder that measures positive vs. negative mood and Google-Profile of Mood States (GPOMS) that measures mood in terms of 6 dimensions (Calm, Alert, Sure, Vital, Kind, and Happy). They cross-validated the resulting mood time series by comparing their ability to detect the public's response to the presidential election and Thanksgiving Day in 2008. A Granger causality analysis and a Self-Organizing Fuzzy Neural Network were then used to investigate the hypothesis that public mood states, as measured by the OpinionFinder and GPOMS mood time series, were predictive of changes in DJIA closing values. The results indicated that the accuracy of DJIA predictions can be significantly improved by the inclusion of

specific public mood dimensions but not others. They found an accuracy of 86.7% in predicting the daily up and down changes in the closing values of the DJIA and a reduction of the Mean Average Percentage Error (MAPE) by more than 6%.

Cristian Lumezanu et al. in 2012 studied the tweeting behavior of Twitter propagandists, users who consistently express the same opinion or ideology, focusing on two online communities: the 2010 Nevada senate race and the 2011 debt- ceiling debate. They identified several extreme tweeting patterns that could characterize users who spread propaganda: sending high volumes of tweets over short periods of time, retweeting while publishing little original content, quickly retweeting, and colluding with other, seemingly unrelated, users to send duplicate or near-duplicate messages on the same topic simultaneously. These four features appear to distinguish tweeters who spread propaganda from other more neutral users and could serve as starting point for developing behavioral-based propaganda detection techniques for Twitter.

Opinion Clustering:

Jing Wang et al. in 2012 proposed the problem of identifying the diversionary comments under political blog posts. The paper showed the categorization of diversionary comments under 5 types and proposed an effective technique to rank comments in descending order of being diversionary. The evaluation on 2,109 comments

under 20 different blog posts from Digg.com shows that the proposed method achieves the high mean average precision of 92.6%. Sensitivity analysis indicated that the effectiveness of the method is stable under different parameter settings.

Lei Zhang and Bing Liu in 2014 introduced the aspect-based opinion mining method, and discussed the model used for aspect extraction approaches. The paper showed multiple approaches used for topic models like Latent Semantic Allocation (LDA) and Multi-grain LDA. For evaluation they used measures for information extraction such as precision, recall and F-1 scores which are also often used in aspect and entity extraction.

Janyce Wiebe et al. in 2003 proposed the question of ability to building frameworks of mining perspectives of agents. The paper started by discussing the tasks addressed by the MPQA project. Then the paper described the framework for annotating, learning and using information about perspective. Finally, the paper reported the results of the preliminary annotation study, machine learning experiments, and clustering experiments. In the annotation study, they found that annotators agreed on about 85% of direct expressions of opinion, about 50% of indirect expressions of opinion, and achieved up to 80% kappa agreement on the rhetorical use of perspective. While they did not present the annotation scheme or agreement study in detail, the results demonstrate the feasibility of annotating information about perspective.

CHAPTER 3

METHODOLOGY

Hypothesis & Influence Quantification:

In this chapter we introduce the challenge of measuring the influence or the effect of main stream media on its audience. This readership could be described in different ways using Twitter, which are addressed in this chapter. However, we focus on representing the opinions of Twitter users generally using vector of sentiment that express the bias or neutrality towards multiple different topics. We first start with describing our research questions and hypothesis in comparison with other research works' hypotheses and approaches in quantifying the influence of media, and then we describe the opinion model that we based our analysis and inferences upon.

Research questions:

1) Mass media shaping the audiences' opinions in multiple topics
2) Audience interaction towards information transmitted with the personal influence arising from social NWs

Often, media users may find themselves in disagreement with certain perspectives uncovered in media content. When that occurs, those with oppositional readings to media turn to other sources to find perspectives that align better with their own (Festinger

1957). Individuals with particularly high levels of disassociation with the media will frequently experience feelings of dissonance (D'Alessio & Allen 2002). These people then make individual media selections that align with their own views and support their own perspectives. Therefore, on the individual level, acceptance of media messages can often be refuted or assimilated within previously held beliefs and not immediately accepted as part of one's own reality. This does not refute the systemic ideological biases embedded within all media (Herman & Chomsky 1988). At the macro-level, one can see ideological consistency throughout society and across media outlets. Quantifying the influence of mass media through Twitter could help us find the factor at which the society relies on news outlets without evaluating the content before agreeing with it. Thus our research question concludes into whether if the mass media shape individuals opinions? And how does the audience interact towards the information transmitted with the personal influence arising from social media?

Other Approaches:

In this subsection we mention three other approaches for solving the research question proposed earlier:

Zhongyu Wei et al. in 2013 analyzed the behavior of mainstream media on Twitter and studied how they exert their influence to shape the public opinion. The hypothesis of this question is that Twitter gives the brief picture about the basic ecology habit of mass media in influencing public opinion. The paper considered three questions

to answer, which are how to quantify bias on Twitter? How information originated from media propagates on Twitter? And how mass media compares with the most influential individuals in terms of social influence? The method was applied on a Twitter dataset collected about the UK general elections, were three major parties played a role. To answer those questions the paper proposed an empirical measure to quantify media bias based on sentiment analysis. First, they try traditional lexicon-based sentiment analysis methods, which failed, since more than 61% of the tweets contain sentiment about more than one party. Thus they used OpenAmplify for entity-level sentiment extraction from tweets. The results showed 54% accuracy when using the traditional lexicon-based sentiment analysis, while 74% when using OpenAmplify. The quantified media bias measure in this paper is represented by the following equation:

$$Media\ Bias_{ij} = \frac{C_{ij}^{pos} + 1}{C_{ij}^{neg} + 1} - 1$$

Where C_{ij}^{pos} and C_{ij}^{neg} denotes the total number of positive and negative tweets from a media outlet i towards a party j. Media Bias takes value 0 if there is no bias. And it is positive for positive bias and negative vice versa.

Then the paper transitioned to the analysis of media intermediates by studying the information propagation. The information propagation is addressed as the retweets which are used to replicate information from news Twitter pages. The intermediates are defined as the direct re-tweeters, and their contribution is measured by several categories, for

example, the retweet rate, the average retweet times per tweet and the life span. Those measures are applied to compare between multiple categories of intermediates like celebrities, bloggers, mainstream media and journalists. Similarly, [60] presented a measure for the tweeting behavior of propagandists on Twitter, and showed the effects through retweets.

Lastly, the paper compared the information diffusion patterns from different categories of sources. Supposing a single information cascade is generated by seed tweet followed by all of its retweets, they calculated the distribution of information cascades by source category, and the observation is that most information cascades are originated from media (including mainstream media and social media) and party.

The second approach introduced by Seth Myers et al. in 2012 focused on both internal and external influence on social networks. In their model they distinguished between exposures and infections. An exposure event occurs when a node gets exposed to information, and an infection event occurs when a node reposts a tweet with the same information. Exposures to information lead to an infection. They developed an estimation technique from a given network and a set of node infection times. The event profile is defined as the user that absorbs external information to the rest of the nodes. The event profiles quantify the number of exposures generated by the external source over time. Additionally, they infer the exposure curve that models the probability of infection as a function of the number of exposures of a node. They experimented with their model on

Twitter and found that the occurrence external out-of-network events are detected accurately, and the exposure curve inferred from the model is often 50% more accurate than baseline methods. However the model was fitted to thousands of different URL's that have appeared across Twitter users, and used the inferred parameters of the model to provide insights into the mechanics of the emergence of these URLs.

The third approach is introduced by DeMarzo et al. in 2003, which proposed a boundedly rational model of opinion formation in which individuals are subject to persuasion bias; that is, they fail to account for possible repetition in the information they receive. They showed that persuasion bias implies the phenomenon of social influence, whereby one's influence on group opinions depends not only on accuracy, but also on how well-connected one is in the social network that determines communication. Persuasion bias also implies the phenomenon of *unidimensional* opinions; that is, individuals' opinions over a multidimensional set of issues converge to a single "left-right" spectrum. They explored the implications of their model in several natural settings, including political science and marketing, and obtained a number of novel empirical implications.

Targeted audience:

Similarly as Seth Myers et al. we distinguish between exposures and infections. Unlike Seth Myers et al. and DeMarzo et al. we disregard internal infections, which mean that our main focus is on analyzing external influences only. When a node U gets exposed to

or becomes aware of information I whenever one of its neighbors in the social network posts a tweet containing I (we call this an internal exposure). However, we consider internal exposures, since the task of distinguishing between internal exposures and infections is a very challenging problem. From our results, we observed another category of users which depends on each news outlet separately. This category concerns news channel referrers and non-referrers. For example, news referrers of Fox news are the users who mentioned Fox news whether using hashtags or without.

Model:

In our approach we model the opinion of Twitter users subjected to persuasion bias from mass media, unlike DeMarzo et al., their model tests the persuasion bias internally. Thus we are concerned about the phenomena of *unidimensional* opinions in afore mentioned paragraph to be the basic measure of influence. Our hypothesis is that blind (loyal) followers to a particular news channel fall into the same herd of opinions and express their *unidimensional* opinion. One of the main features that differentiate *unidimensional* opinions from other diverse perspectives is the isolation property. According to such assumption we defined the main task is to detect isolated opinions on multiple issues (topics). Then we quantify the assurance factor of influence of a particular channel as the percentage of tweets which referred that news channels out of the total

number of isolated tweets. We assume that news channels referred in a tweet is the source of information that resulted in biasing the opinion of that tweet.

Our aspect-based opinion mining framework is based on modelling opinions into vectors of sentiment towards different topics T_j. An opinion O_i expressed in a $tweet_i$ using the sentiment based assignment values S_T for each of the topics from $T_1$to T_n follow the vector representation below:

$$O_i = \{S_1, S_2, \dots, S_n\}$$

Sentiment values depend on the method on which we categorize the sentiment, which will be mentioned in more details in the next section (i.e. scoring, groups, trivial polarity). However, each method uses one of the categories at a time. An opinion group O_g is a set of combinations of sentiment vectors that are very similar to each other. Those groups of opinions are clustered using Expectation Maximization (EM) algorithm.

The problem of recognizing blind followers relies on detecting which group of clustered opinions is isolated from the rest of the clusters. Thus, we are looking at the distribution of clusters among the sentiment towards each topic, while considering the number of referrers that are in the isolated cluster. One of the main advantages of using EM algorithm is that the results indicate the mean and the standard deviation of the clusters towards each attribute, which is the topic in our context. An isolated cluster is defined as the cluster that has no other overlapping clusters in terms of the sentiment

values that it spans to a certain topic. The isolated clusters are defined as the ones that do not overlap with other clusters. By this definition we can calculate the minimum and maximum of each cluster using the mean and standard deviation resulting from EM algorithm, then find overlapping and non-overlapping clusters. Consequently, the non-overlapping clusters are the isolated ones. To understand the importance of detecting isolated opinion groups, we show the resulting visuals of the EM algorithm. The visuals contribute to show the other point of view to the isolation property of opinion groups, which is diversity. Diversity is claimed for a certain range of sentiment values towards a topic, where this range should contain more than one opinion group if it is diverse. However, the diversity cannot be quantified, only through the negation effect of isolation.

Framework of the aspect-based opinion mining process:

In this section we reveal the framework of algorithms and techniques used to mine the opinions of Twitter users towards multiple the trending topics, inside the collected dataset. We describe in details the languages and tools used and all technical difficulties faced through the project. The framework is composed of three steps:

- Trending Topics extraction
- Sentiment Analysis
- Opinion clustering

As shown in figure 1, we first start with mining the trending topics using Apriori algorithm from two different inputs, the hashtags and the most frequent words. The elements in white circles are the optional inputs which could be provided to the step, where it refers to, which means that either of the inputs is experimented one at a time. The difference between using both inputs is explained in the results chapter. The output of the Apriori algorithm is the frequent itemsets, where each word is an item representing a topic that concerns the users. The second step is calculating and assigning the sentiment to construct the sentiment matrix, which the clustering process is based on. The sentiment values used are categorized into three; trivial polarity, adjective hierarchy and scoring, where each category resulted into different number, distribution and output formats of clusters. The sentiment categories are explained in details in the next section, and the resulted clusters from each category are discussed in the results chapter.

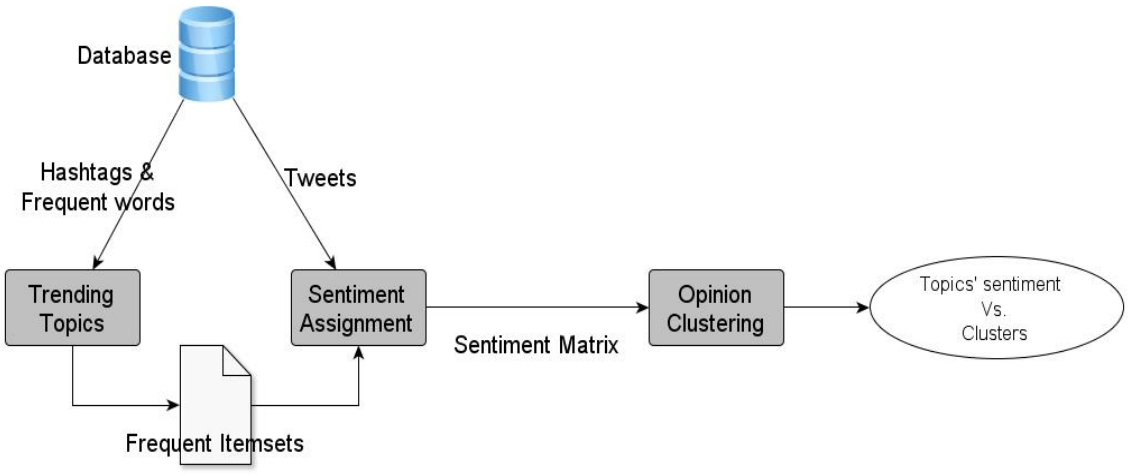

Figure 1. The framework of the aspect-based opinion mining process utilizes those three main steps: Tredngin Topics, Sentiment Assignment and Opinion Clustering.

To our knowledge this framework has not been investigated by any research work before, and the validation proves the compliance to the hypothesis mentioned with those steps. The data collection, the different analysis methods and their results are discussed in the next chapter (Results & Discussion).

Trending Topics extraction (Apriori):

In this section we cast the challenge of finding the frequent itemset problem as the trending topics by the dataset collected through keywords. Although, one can think that by default that the harvesting keywords used in streaming the tweets will be mostly the dominant factor and similar to the output of frequent itemsets, the results show that it is not totally true. Here we describe the Apriori algorithm, which was used to find the trending topics in the collected tweets. We conducted two main experiments to mine the trending topics. In the first we used the most frequent words as the input but filtering out stop words, while in the second we used all hashtags instead. The results are explained in the next chapter to fill out the reasoning of which method is better (Latiri et al. 2001).

Apriori:

In the case of an indexed tweet collection, the indexing structures (keyword sets) can be used as a basis for information extraction. In such a framework, one possible goal is to extract significant keyword associations. Let's consider a set of key- words $A = \{w_1, w_2, \ldots, w_m\}$ and a collection of indexed tweets $T = \{t_1, t_2, \ldots, t_n\}$ (i.e. each t_i is

associated with a subset of A denoted $t_i(A)$). Let $W \subseteq A$ be a set of key-words, the set of all tweets t_i in T such that $W \subseteq t(A)$ will be called the covering set for W and denoted $[W]$. Any pair (W, w), where $W \subseteq A$ is a set of keywords and $w \in A/W$, will be called an association rule (or simply an association), and denoted $W \Rightarrow w$.

Given an association rule R: $(W \Rightarrow w)$,

- $S(R,T) = |[W \cup \{w\}]|$ is called the support of R with respect to the collection T ($|X|$ $denotes\ the\ size\ of\ the\ set\ X$)
- $C(R,T) = \frac{|[W \cup \{w\}]|}{|[W]|}$ is called the confidence of R with respect to the collection T

Notice that $C(R,T)$ is an approximation (maximum likelihood estimate) of the conditional probability for a text of being indexed by the keyword w if it is already indexed by the key-word set W.

An association rule R generated from a collection of texts T is said to satisfy support and confidence constraints σ and γ if

$S(R,T) \geq \sigma$ And $C(R,T) \geq \gamma$

To simplify notations, $[W \cup \{w\}]$ will be often written $[W\ w]$ and a rule R: $(W \Rightarrow w)$ satisfying given support and confidence constraints will be simply written as:

$W \Rightarrow w$, where $S(R,T)/C(R,T)$

Informally, for an association rule ($W \Rightarrow w$), such σ/γ constraints can be interpreted as: there exists a significant number of tweets (at least σ), for which being related to the topic characterized by the keyword set W implies (with a conditional probability estimated by γ) to be also related to the topic characterized by the keyword w.

As far as the actual association extraction is concerned, the common procedures are usually two steps algorithms:

- Generation of all the keywords sets with support at least equal to σ (i.e. all the keywords sets Wsuch that $|[W]| \geq \sigma$). The generated keywords sets are called the frequent sets (or σcovers)
- Generation of all the association rules that can be derived from the produced frequent sets and that satisfy the confidence constraint γ

The frequent sets are obtained by incremental algorithms that explore the possible keywords subsets, starting from the frequent singletons (i.e. the $\{w\}$ such that $|[\{w\}]| \geq \sigma$) and iteratively adding only those keywords that produce new frequent sets. This step is the most computationally expensive (exponential in the worst case) in the extraction procedure.

The associations derived from a frequent set W are then obtained by generating all the implications of the form $W/\{w\} \Rightarrow w$, $(w \in W)$, and keeping only the ones satisfying

the confidence constraint γ. Some additional treatment (structural or statistical pruning, redundancy elimination) is usually added to the extraction procedure in order to reduce the number of generated associations. Nevertheless, we did not consider the second step in finding the association rules, since we are looking for frequent sets only (Chengqi Zhang, Shichao Zhang et al. 2002).

On the implementation side, figure 2 shows the generic view of the incremental procedure for finding the candidate itemsets and the frequent itemsets.

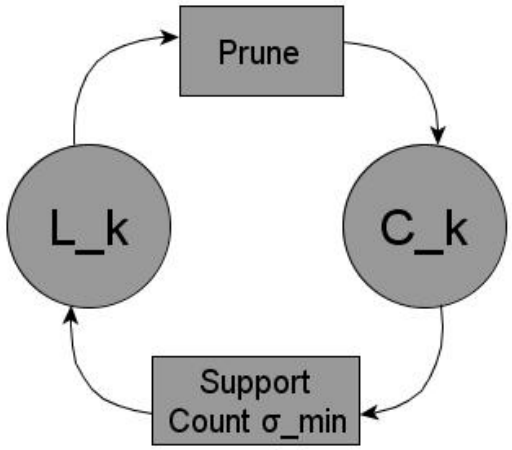

Figure 2. The two alternating steps of the Apriori algorithm between pruning and support count filtering.

The following steps intend to show the pseudo-code which was implemented in C++ on fedora for conducting our experiments described in the next chapter:

C_k: Candidate itemset of size k

L_k: Frequent itemset k

$L_1 = \{frequent\ items\}$;

for ($k = 1$; $L_k! = \emptyset$; $k++$) do begin

C_{k+1} =candidates generated from L_k

for each tweet t_i in the database do

increment the count of all candidates in C_{k+1}that are contained in t_i

L_{k+1} =candiates in C_{k+1} with support count $\geq \sigma_{min}$

end

return $\bigcup_k L_k$;

end

It is very important to demonstrate the pruning step, since it reduces the memory space consumed between each incremental step and heavy computation due to large C_{k+1} generated. A next candidate C_{k+1} is said to satisfy the pruning condition, when all its subsets are present in the frequent itemset L_k. For example, a candidate $\{A, B, C\}$ passes the pruning step if and only if $\{A, B\}$, $\{A, C\}$ and $\{B, C\}$ are present in the frequent itemset. The following example is intended to show the whole process of pruning the frequent itemsets using their subsets and filtering the candidate itemsets using σ_{min}.

Consider the a database consisting of 9 tweets in the table below, and suppose the $\sigma_{min} = 22\%$, which means 2 out of the 9 tweets. The items are numbered with a prefix I.

Table 1.1 Transactions of tweets example. Example of tweets contain different combinations of keywords

Tweet ID	List of keywords
1	$l1, l2, l5$
2	$l2, l4$
3	$l2, l3$
4	$l1, l2, l4$
5	$l1, l3$
6	$l2, l3$
7	$l1, l3$
8	$l1, l2, l3, l5$
9	$l1, l2, l3$

The first step is to generate the 1-itemset frequent pattern, which can be found by counting the frequency of each item individually.

Table 1.2 Following the example, the support counts of 1-itemsets according to table 1.1

Itemsets L_1	Support Counts

$\boldsymbol{l1}$	6
$\boldsymbol{l2}$	7
$\boldsymbol{l3}$	6
$\boldsymbol{l4}$	2
$\boldsymbol{l5}$	2

It appears that all candidates satisfy the σ_{min} of 22% specified previously. Now it is time to generate the 2-itemset candidate pattern, which is the following table, with their support counts:

Table 1.3 The support counts of 2-temsets according to table 1.1

Itemsets L_2	**Support Counts**
$\boldsymbol{l1, l2}$	4
$\boldsymbol{l1, l3}$	4
$\boldsymbol{l1, l4}$	1
$\boldsymbol{l1, l5}$	2
$\boldsymbol{l2, l3}$	4
$\boldsymbol{l2, l4}$	2
$\boldsymbol{l2, l5}$	2
$\boldsymbol{l3, l4}$	0
$\boldsymbol{l3, l5}$	1
$\boldsymbol{l4, l5}$	0

Although, this is the second least candidate pattern in number of items, it contains the largest number of itemsets possibilities in comparison with other n-itemsets candidate patterns. Thus the advantage of allocating memory incrementally is appreciated when pruning is applied. And by applying the filter of minimum support count the following is the 2-itemset frequent pattern:

Table 1.4 The support counts of 2-frequent itemsets after filtering according to the minimum support counts.

Itemsets L_2	**Support Counts**
$\boldsymbol{l1, l2}$	4
$\boldsymbol{l1, l3}$	4
$\boldsymbol{l1, l5}$	2
$\boldsymbol{l2, l3}$	4
$\boldsymbol{l2, l4}$	2
$\boldsymbol{l2, l5}$	2

Till now we have not used the Apriori property yet, since the pruning effect has not been applied. It will be more obvious now when generating the 3-itemsets candidate pattern. Transitioning to C_3 requires the initial suggested candidates which requires joining the items as following:

$$C_3 = \{\{l1, l2, l3\}, \{l1, l2, l5\}, \{l1, l3, l5\}, \{l2, l3, l4\}, \{l2, l3, l5\}, \{l2, l4, l5\}\}$$

For example, $\{l1, l2, l3\}$, the 2-item subsets of it are $\{l1, l2\}$, $\{l2, l3\}$ and $\{l1, l3\}$. Since all 2-item subsets of $\{l1, l2, l3\}$are members of L_2, we will keep $\{l1, l2, l3\}$ in C_3. Another contrary example, $\{l2, l3, l5\}$ which shows how the pruning is performed. The 2-item subsets are $\{l2, l3\}$, $\{l2, l5\}$ and $\{l3, l5\}$, but $\{l3, l5\}$ is not a member in L_2 and hence it is not frequent, violating the Apriori property. Thus we will remove the $\{l1, l2, l3\}$ from C_3. Therefore, $C_3 = \{\{l1, l2, l3\}, \{l1, l2, l5\}\}$, which satisfy the minimum support count to be the L_3. Finally, when transitioning to the 4-itemset candidate pattern the join operation on L_3fails to generate any itemset for $C_4 = \emptyset$. The algorithm terminates, having found all of the frequent itemsets.

The last step is generating the association rules from the frequent itemsets resulted. However, we did not use the association rules to represent the trending topics; we only used those important words that were inside different sizes of the frequent itemsets. For each frequent itemset L, all nonempty subsets s of L are generated. Then for every nonempty subset $s$of L, an output rule is "$s \Rightarrow L - s$" if $\frac{supportCount(L)}{supportCount(s)} \geq \gamma_{min}$. Using the same example if we took $\{l1, l2, l5\}$, all its nonempty subsets are $\{\{l1, l2\}, \{l1, l5\}, \{l2, l5\}, \{l1\}, \{l2\}, \{l5\}\}$ and $\gamma_{min} = 0.7$. Thus, the selected resulting rules from the table below are the ones above 70%, which have their percentages marked red:

Table 1.5 The confidence level of potential rules, the red marked ones are above the minimum confidence used in this example. The notation of the $sc()$ function means the support count of the itemset between the parentheses.

Rules	Confidence
$\{\boldsymbol{l1}, \boldsymbol{l2}\} \Rightarrow \boldsymbol{l5}$	$\frac{sc(\{l1, l2, l5\})}{sc(\{l1, l2\})} = \frac{2}{4} = 50\%$
$\{\boldsymbol{l1}, \boldsymbol{l5}\} \Rightarrow \boldsymbol{l2}$	$\frac{sc(\{l1, l2, l5\})}{sc(\{l1, l5\})} = \frac{2}{2} = 100\%$
$\{\boldsymbol{l2}, \boldsymbol{l5}\} \Rightarrow \boldsymbol{l1}$	$\frac{sc(\{l1, l2, l5\})}{sc(\{l2, l5\})} = \frac{2}{2} = 100\%$
$\boldsymbol{l1} \Rightarrow \{\boldsymbol{l2}, \boldsymbol{l5}\}$	$\frac{sc(\{l1, l2, l5\})}{sc(\{l1\})} = \frac{2}{6} = 33\%$
$\boldsymbol{l2} \Rightarrow \{\boldsymbol{l1}, \boldsymbol{l5}\}$	$\frac{sc(\{l1, l2, l5\})}{sc(\{l2\})} = \frac{2}{7} = 29\%$
$\boldsymbol{l5} \Rightarrow \{\boldsymbol{l1}, \boldsymbol{l2}\}$	$\frac{sc(\{l1, l2, l5\})}{sc(\{l5\})} = \frac{2}{2} = 100\%$

There is one last implementation issue that is worth mentioning for memory reduction during the generation of candidate itemsets, which is shown in the code addresses in the appendix A.1. Since the generation of suggestions for candidate itemsets before pruning exponentially consumes the memory, we efficiently implement this step by integrating it with the support count filtering step to test each individual itemset separately then include it in the frequent itemset if satisfies σ_{min}. That means if we have a candidate itemset generated from L_k we pass it individually, without storing it in an actual C_{k+1} of itemsets,

to be tested for pruning. Then if it passed the pruning it is tested for the support count. The cycle is repeated when the itemsets in L_k are all tested and stored in L_{k+1}. The only exceptional step is C_2, since we need to generate all possible combinations between the 1-itemsets frequent patterns. Thus, as clarified by the code comments we separate the steps of generating the 1 & 2-itemsets frequent patterns and the generic number-itemsets frequent patterns.

Sentiment Analysis:

In this step we propose three different approaches in defining the sentiment used then assign for each tweet the appropriate sentiment according to the category defined. The sentiment assignment totally depends on the adjective used in the tweets towards different topics. Nevertheless, we consider only one adjective in the tweet. According to the value and category the adjective falls into, the sentiment assigned only to the topics mentioned in the same tweet, while the rest of the topics are assigned to be neutral (zero). For example, if the adjective was recognized to a corresponding value of x and the only mentioned topics are of index 1, 3 and 4 out of K topics, then the sentiment vector representing this $tweet_i$ will be as following:

$$O_i = \{x, 0, x, x, 0, \dots\}_K$$

In the three methods we used the NLTK[1] platform implemented in python to detect the adjectives in the tweets. NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, and tagging, parsing, and semantic reasoning. WordNet is accessed just another NLTK corpus reader, and can be imported like this:

```
>>> from nltk.corpus import wordnet as wn
```

So now wn in a program would be considered as a variable that contains the WordNet corpus.

The following are the three different methods for assigning the sentiment of the tweets:

1. Trivial polarity
2. Scoring
3. Adjective Hierarchy (semantic relatedness)

Trivial polarity: In this method we downloaded two lists of positive and negative adjectives from []. We developed programs in python to extract the adjectives by tokenizing and then tagging the sentences in the tweets as seen in appendix B.1. The words which match the tag “JJ” are the adjectives, thus we compare those words with

[1] http://nltk.googlecode.com/svn/trunk/doc/howto/wordnet.html

both the positive and negative lists downloaded. If the adjective matches a word in the positive list the nominal value "P" is assigned, while if it matches a word in the negative list the nominal value "N" is assigned, if it did not match any of the lists a nominal value of "N" is assigned. However, some tweets contain more than one adjective, and if both contradict by matching both the positive and negative lists, the nominal value "M" is assigned.

Scoring: In this method we also downloaded a list containing 2,477 adjectives and their scores rated from -5 to +5 by Finn Nielsen in 2009-2011. The list is called "AFINN" and can be downloaded from[2]. This list was used by Lars Kai Hansen et al. in 2011 for sentiment analysis on Twitter. The same process of tokenizing and tagging the sentences takes place in this method too but the adjectives are compared with the scoring list. The score of the adjective is assigned to the topics mentioned in the vector, and if there is more than one adjective in the tweet, the average replaces both scores.

Adjective Hierarchy: In this method we list all adjectives used in the analyzed tweets, also using tokenization and tagging. The goal of listing all the adjectives is to find how they are related semantically using the lexicon imported from WordNet, and then group those words which are the closest to each other as groups of sentiment. Those groups are the basics of sentiment values in this method. The semantic relations give the distance between each adjective and the other through the synonym list. We first look up

[2] http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010

the synonym list of each adjective in the list through the synstes() function. This function has an optional pos argument which lets you constrain the part of speech of the word, for example:

```
>>> wn.synsets('dog')
[Synset('dog.n.01'), Synset('frump.n.01'), Synset('dog.n.03'), Synset('cad.n.01'),
Synset('frank.n.02'), Synset('pawl.n.01'), Synset('andiron.n.01'), Synset('chase.v.01')]
>>> wn.synsets('dog', pos=wn.VERB)
[Synset('chase.v.01')]
```

The other parts of speech are NOUN, ADJ and ADV. A synset is identified with a 3-part name of the form: word.pos.nn. NLTK also facilitates functions to obtain the definition, examples, lemmas and the lemmas' sysnets, as the following functions using the previous example:

```
>>> wn.synset('dog.n.01')
Synset('dog.n.01')
>>> wn.synset('dog.n.01').definition
'a member of the genus Canis (probably descended from the common wolf) that has been
domesticated by man since prehistoric times; occurs in many breeds'
```

```
>>> wn.synset('dog.n.01').examples

['the dog barked all night']

>>> wn.synset('dog.n.01').lemmas

[Lemma('dog.n.01.dog'), Lemma('dog.n.01.domestic_dog'),
Lemma('dog.n.01.Canis_familiaris')]

>>> [lemma.name for lemma in wn.synset('dog.n.01').lemmas]

['dog', 'domestic_dog', 'Canis_familiaris']

>>> wn.lemma('dog.n.01.dog').synset

Synset('dog.n.01')
```

Synets by the NLTK definition is a set of synonyms that share a common meaning. Each synset contains one or more lemmas, which represent a specific sense of a specific word. For example:

```
>>> dog = wn.synset('dog.n.01')

>>> dog.hypernyms()

[Synset('domestic_animal.n.01'), Synset('canine.n.02')]

>>> dog.hyponyms()
```

[Synset('puppy.n.01'), Synset('great_pyrenees.n.01'), Synset('basenji.n.01'), ...]

>>> dog.member_holonyms()

[Synset('pack.n.06'), Synset('canis.n.01')]

>>> dog.root_hypernyms()

[Synset('entity.n.01')]

Thus we give the following definitions from[3] as a reference for the reader to interpret the linguistic meaning of:

Synonyms: are words with the same or similar meanings.

Antonyms: a word opposite in meaning to another. Fast is an antonym of slow.

Hypernym: A linguistic term for a word whose meaning includes the meanings of other words. For instance, flower is a hypernym of daisy and rose.

Hyponym: In linguistics, a specific term used to designate a member of a class. For instance, daisy and rose are hyponyms of flower.

Holonyms: A term that denotes a whole whose part is denoted by another term, such as 'face' in relation to 'eye'.[4]

[3] http://grammar.about.com/
[4] http://en.wiktionary.org/wiki/

Pertainyms: (computational linguistics) a word, usually an adjective, which can be defined as "of or pertaining to" another word.

However, some relations have to be defined by WordNet only over Lemmas (i.e. antonyms, derivationally related forms and pertainyms). The following example shows how they can be obtained:

```
>>> eat = wn.lemma('eat.v.03.eat')

>>> eat

Lemma('feed.v.06.eat')
```

Where Lemmas can also have relations between them, which can only apply on Lemmas not on synsets for example:

```
>>> vocal = wn.lemma('vocal.a.01.vocal')

>>> vocal.derivationally_related_forms()

[Lemma('vocalize.v.02.vocalize')]

>>> vocal.pertainyms()

[Lemma('voice.n.02.voice')]

>>> vocal.antonyms()

[Lemma('instrumental.a.01.instrumental')]
```

At the end we only used the sysnset() function of the adjectives without restricting a pos argument to them in order to calculate the score of the similarity between their each other's senses. There are multiple ways to calculate this score that denotes how two similar word senses are.

First the synonym lists are retrieved for each adjective using the synset() function. For example, if we would like to find the semantic relation between the words 'dog' and 'cat':

```
>>> dog = wn.synset('dog.n.01')

>>> cat = wn.synset('cat.n.01')
```

Using NLTK we have three options for denoting the similarity between both words:

Path similarity: using the function synset1.path_similarity(synset2)

The function returns a score denoting how similar two word senses are, based on the shortest path that connects the senses in the is-a (hypernym/hypnoym) taxonomy. The score is in the range 0 to 1, except in those cases where a path cannot be found (will only be true for verbs as there are many distinct verb taxonomies), in which case -1 is returned. A score of 1 represents identity i.e. comparing a sense with itself will return 1. For example:

```
>>> dog.path_similarity(cat)
```

0.20000000000000001

Leacock-Chodorow Similarity: using synset1.lch_similarity(synset2)

The function returns a score denoting how similar two word senses are, based on the shortest path that connects the senses (as above) and the maximum depth of the taxonomy in which the senses occur. The relationship is given as -log(p/2d) where p is the shortest path length and d the taxonomy depth. For example:

>>> dog.lch_similarity(cat)

2.0281482472922856

Wu-Palmer Similarity: using synset1.wup_similarity(synset2)

The function returns a score denoting how similar two word senses are, based on the depth of the two senses in the taxonomy and that of their Least Common Subsumer (LCS) (most specific ancestor node). Note that at this time the scores given do not always agree with those given by Pedersen's Perl implementation of WordNet Similarity. The LCS does not necessarily feature in the shortest path connecting the two senses, as it is by definition the common ancestor deepest in the taxonomy, not closest to the two senses. Typically, however, it will so feature. Where multiple candidates for the LCS exist that whose shortest path to the root node is the longest will be selected. Where the LCS has multiple paths to the root, the longer path is used for the purposes of the calculation. For example:

```
>>> dog.wup_similarity(cat)

0.8571428571428571
```

Additionally, we can use another three functions when defining the information content dictionary. Information Content (IC): loads an information content file from the wordnet_ic corpus, where we can also specify the information content of certain lists to be held in variables, for example:

```
>>> from nltk.corpus import wordnet_ic

>>> brown_ic = wordnet_ic.ic('ic-brown.dat')

>>> semcor_ic = wordnet_ic.ic('ic-semcor.dat')
```

Moreover, there is an option to create an information content dictionary from a corpus (or anything that has a words() method). Using the following example:

```
>>> from nltk.corpus import genesis

>>> genesis_ic = wn.ic(genesis, False, 0.0)
```

The three methods are:

Resnik Similarity: using synset1.res_similarity(synset2, ic)

The function returns a score denoting how similar two word senses are, based on the IC of the LCS (most specific ancestor node). Note that for any similarity measure that

uses information content, the result is dependent on the corpus used to generate the information content and the specifics of how the information content was created. For example:

```
>>> dog.res_similarity(cat, brown_ic)

7.9116665090365768

>>> dog.res_similarity(cat, genesis_ic)

7.1388833044805002
```

Jiang-Conrath Similarity: using synset1.jcn_similarity(synset2, ic)

The function returns a score denoting how similar two word senses are, based on the Information Content (IC) of the Least Common Subsumer (most specific ancestor node) and that of the two input Synsets. The relationship is given by the equation 1 / (IC(s1) + IC(s2) - 2 * IC(lcs)). For example:

```
>>> dog.jcn_similarity(cat, brown_ic)

0.44977552855167391

>>> dog.jcn_similarity(cat, genesis_ic)

0.28539390848096979
```

Lin Similarity: using synset1.lin_similarity(synset2, ic)

The function returns a score denoting how similar two word senses are, based on the Information Content (IC) of the Least Common Subsumer (most specific ancestor node) and that of the two input Synsets. The relationship is given by the equation 2 * IC(lcs) / (IC(s1) + IC(s2)). For example:

>>> dog.lin_similarity(cat, semcor_ic)

0.88632886280862277

We used the Wu-Palmer similarity since it features the common ancestor deepest in the taxonomy not closest to the two senses. The program written in python, appendix B.1, collects all the adjectives in the tweets and calculates the distance matrix in terms of Wu-Palmer similarity. Finding the similarity is based on the SemCor corpus which is a subset of the Brown corpus. SemCor corpus is a sense-tagged corpora created at Princeton University by the WordNet Project research team[5], which defines the relational taxonomy between words. The reason for using the SemCor corpus is that it has the highest percentage of adjective connections. The distance matrix then is used to construct the hierarchy of the adjectives within the list. By this hierarchy we grouped the adjectives as the sentiment values, so the sentiment values of the tweet will depend on adjective choice that was used from those groups. We used the R programming language to apply the hierarchical clustering algorithm, and the input and output formats and the functions are explained in this section too.

[5] http://www.gabormelli.com/RKB/SemCor_Corpus

Hierarchical Clustering Algorithm:

Hierarchical clustering algorithm is used in many data mining applications to build a binary tree of data that successively merges similar groups of points. Visualizing such information provides useful summary of the data, but we used this type of tree, which is called “Dendogram”, in our analysis to define a threshold separating the adjectives into groups of sentiment values. This separation could be defined number of groups or level based. The algorithm only requires a measure of similarity or dissimilarity between groups of data points. At first each point could be viewed as an entity group by itself, then the algorithm decides to merge pairs of these groups incrementally until all of the data points are one single group. This type of hierarchical clustering is called “Agglomerative”. While if all data points at first are considered as a single group then algorithm works the opposite way by splitting up this group into pairs incrementally, then it is said to be “Divisive”.

There are several types of metrics that can be used, which are basically the formula on which the distance matrix was built upon. For example, the Euclidean distance squared Euclidean distance, Manhattan distance, maximum distance, Mahalanobis distance, cosine similarity, Hamming distance and Levenshtein distance. Although, all of these metrics are the standards used in most of the applications, the most appropriate metric is based on the scoring that denotes the similarity between word

senses. We convert this similarity into dissimilarity matrix by the similarity score from one, since the maximum score is one. The reason for using dissimilarity matrix is that most of the free software (i.e. R and Weka) available now has the standard of using it instead of the similarity matrix, except if it is an option to change. The following are the formulas for the standard metric criteria that can be used:

$$Eucledian\ distance: \left|\left|a-b\right|\right|_2 = \sqrt{\sum_i (a_i - b_i)^2}$$

$$\text{squared Euclidean distance: } \left|\left|\text{a}-\text{b}\right|\right|_2^2 = \sum_i (a_i - b_i)^2$$

$$Manhattan\ distance: \left|\left|a-b\right|\right|_1 = \sum_i |a_i - b_i|$$

$$Maximum\ distance: \left|\left|a-b\right|\right|_\infty = max_i|a_i - b_i|$$

$$Mahalanobis\ distance: \sqrt{(a-b)^T S^{-1}(a-b)}, where\ S\ is\ the\ covariance\ matrix$$

$$Cosine\ similarity: \frac{ab}{\left|\left|a\right|\right|\left|\left|b\right|\right|}$$

Another feature in the hierarchical clustering algorithm that should be specified when using is the linkage criteria. The linkage criterion determines the distance between sets of observations as a function of the pairwise distances between observations. Some

commonly used linkage criteria between two sets of observations A and B, where d is the chosen metric, are (SAS/STAT 9.2 Users Guide):

$$Maximum\ or\ complete\ linkage\ clustering: \max\{d(a,b): a \in A, b \in B\}$$

$$Minimum\ or\ single-linkage\ clustering: \min\{d(a,b): a \in A, b \in B\}$$

$$Mean\ or\ average\ linkage\ clustering, or\ UPGMA: \frac{1}{|A|\,|B|}\sum_{a\in A}\sum_{b\in B} d(a,b)$$

$$Minimum\ energy\ clustering: \frac{2}{nm}\sum_{i,j=1}^{n,m}||a_i - b_i||_2 - \frac{1}{n^2}\sum_{i,j=1}^{n}||a_i - a_j||_2 - \frac{1}{m^2}\sum_{i,j=1}^{m}||b_i - b_j||_2$$

In order to demonstrate the process of the hierarchical clustering algorithm, we investigate the example of clustering some of Italian cities by distance in kilometers using the single-linkage criteria. The input is simple as the following table:

Table 2.1 The distance matrix between Italian cities as an input for the Hierarchical clustering algorithm.

	BA	**FI**	**MI**	**NA**	**RM**	**TO**
BA	0	662	877	255	412	996

FI	662	0	295	468	268	400
MI	877	295	0	754	564	138
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0

Apparently, the distance matrix is replicated on both sides of the diagonal, which is an advantage in our case that we utilized to reduce complexity by half when calculating the dissimilarity matrix between adjectives.

The nearest pair of cities is MI and TO, at distance 138. These are merged into a single cluster called "MI/TO". The level of the new cluster is $L(MI/TO) = 138$.

Then we compute the distance from this new compound object to all other objects. In single link clustering the rule is that the distance from the compound object to another object is equal to the shortest distance from any member of the cluster to the outside object. So the distance from "MI/TO" to RM is chosen to be 564, which is the distance from MI to RM, and so on. After merging MI with TO we obtain the following matrix:

Table 2.2 The distance matrix after the first step in merging the closest two objects (cities); MI and TO.

	BA	FI	MI/TO	NA	RM
BA	0	662	877	255	412
FI	662	0	295	468	268
MI/TO	877	295	0	754	564
NA	255	468	754	0	219
RM	412	268	564	219	0

$min\ d(i,j) = d(NA, RM) = 219 \Rightarrow\ merge\ NA\ \&\ RM$ into a new cluster called NA/RM;

$$L\left(\frac{NA}{RM}\right) = 219$$

Table 2.3 The distance after merging the closest two objects (two groups of cities); Na and RM, according to the previous distance previous matrix in table 2.2

	BA	FI	MI/TO	NA/RM
BA	0	662	877	255
FI	662	0	295	268
MI/TO	877	295	0	564
NA/RM	255	268	564	0

$min\ d(i,j) = d\left(BA, \frac{NA}{RM}\right) = 255\ \Rightarrow\ merge\ BA\ \&\ NA/RM$ into a new cluster called $BA/NA/RM$

$$L\left(\frac{\frac{BA}{NA}}{RM}\right) = 255$$

Table 2.4 The distance after merging the closest two objects (two groups of cities); BA and Na/RM, according to the previous distance previous matrix in table 2.3.

	BA/NA/RM	**FI**	**MI/TO**
BA/NA/RM	0	268	564
FI	268	0	295
MI/TO	564	295	0

$min\ d(i,j) = d(BA/NA/RM, FI) = 268 \Rightarrow merge\ BA/NA/RM\ \&\ FI$ into a new cluster called $BA/FI/NA/RM$

$$L(BA/FI/NA/RM) = 268$$

Table 2.5 The distance after merging the closest two objects (two groups of cities); FI and BA/Na/RM, according to the previous distance previous matrix in table 2.4.

	BA/FI/NA/RM	**MI/TO**
BA/FI/NA/RM	0	295
MI/TO	295	0

Finally, we merge the last two clusters at level 295, and the process is summarized by the following hierarchical tree (Dendogram), where we actually see how the cities merge at different heights:

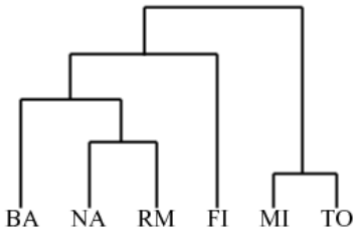

Figure 3 The Dendogram of the relations between the Italian cities according to their location. The Hierarchical structure summarizes the steps of merging throughout the process.

Thus, the cities can be grouped using a certain value of level or by specifying the number of groups that needs to be formed from the Dendogram.

Hierarchical clustering using R programming:

The R programming software is available online for free, which is used by many analysts in the industry, due to its ease-of-use and portability on various types of machines (i.e. OSX, Windows, Linux). It is installed on our fedora machine at CAU. Our concern is to use the hierarchical clustering algorithm to find the semantic relation

between the adjectives used in the tweets collected and build a Dendogram of how those adjectives could be grouped. The algorithm is implemented using the method[6]:

$$hclust()$$

And the possible arguments which the method receives are presented in the following table:

Table 3.1 Shows the possible arguments of the hclust() function in R programming and their descriptions.

Argument	Description
D	A dissimilarity structure as produced by dist
Method	The agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward", "single", "complete", "average", "mcquitty", "median" or "centroid"
Members	NULL or a vector with length size of d. See the 'Details' section
x, tree	an object of the type produced by hclust
Hang	The fraction of the plot height by which labels should hang below the rest of the plot. A negative value will cause the labels to hang down from 0
Labels	A character vector of labels for the leaves of the tree. By default the row names or row numbers of the original data are used. If labels = FALSE no labels at all are plotted
axes, frame.plot, ann	logical flags as in plot.default
main, sub, xlab, ylab	character strings for title. sub and xlab have a non-NULL default when there's a tree$call
...	Further graphical arguments

[6] http://stat.ethz.ch/R-manual/R-patched/library/stats/html/hclust.html

Unit	logical. If true, the splits are plotted at equally-spaced heights rather than at the height in the object
Hmin	All heights less than hmin are regarded as being hmin: this can be used to suppress detail at the bottom of the tree (numeric values)
level, square, plot	as yet unimplemented arguments of plclust for S-PLUS compatibility

This function performs a hierarchical cluster analysis using a set of dissimilarities for the n objects being clustered. Initially, each object is assigned to its own cluster and then the algorithm proceeds iteratively, at each stage joining the two most similar clusters, continuing until there is just a single cluster. At each stage distances between clusters are recomputed by the Lance–Williams dissimilarity update formula according to the particular clustering method being used. Thus, we follow the steps of scanning the distance matrix and converting it to a distance object representing all adjectives as separate objects to build the Dendogram upon. As shown in appendix B.2 we follow these steps to divide the adjectives into groups through the hierarchical structure created from their semantic relatedness:

1. Scan the lower the file of the distance matrix
2. Calculate the number of columns of the matrix
3. Create an empty matrix with the number of rows and columns as the number calculated
4. Scan the file into the matrix created

5. Transpose the matrix
6. Row and column bind the matrix
7. Convert the distance matrix into a distance object
8. Execute the agglomerative hierarchical clustering method from the distance object using single linkage method
9. Cut the tree to create five separate groups of sentiment
10. Write a file containing each adjective and its corresponding group
11. Plot the tree (Dendogram)

The program in appendix B.2 shows how we used R programming in clustering the adjectives into groups by applying the hierarchical clustering algorithm implemented in R. The script first collects the tweets then extracts all the adjectives using NLTK to put them in a list. This list is used to find the distance matrix between each adjective and the other. Lastly, the R script scans this file of distance matrix to convert it into a distance object for the $hclust$ function as shown in the steps above.

However, there is a number of different clustering methods are provided. Ward's minimum variance method aims at finding compact, spherical clusters. The complete linkage method finds similar clusters. The single linkage method (which is closely related to the minimal spanning tree) adopts a ‘friends of friends’ clustering strategy. The other methods can be regarded as aiming for clusters with characteristics somewhere between the single and complete link methods. Note however, that methods "median" and

"centroid" are not leading to a monotone distance measure, or equivalently the resulting Dendrograms can have so called inversions (which are hard to interpret).

If the argument $members != NULL$, then d is taken to be a dissimilarity matrix between clusters instead of dissimilarities between singletons and members gives the number of observations per cluster. This way the hierarchical cluster algorithm can be started in the middle of the Dendrogram, e.g., in order to reconstruct the part of the tree above a cut. Therefore, we consider members as NULL value. Dissimilarities between clusters can be efficiently computed (i.e., without hclust itself) only for a limited number of distance/linkage combinations, the simplest one being squared Euclidean distance and centroid linkage. In this case the dissimilarities between the clusters are the squared Euclidean distances between cluster means.

In hierarchical cluster displays, a decision is needed at each merge to specify which sub-tree should go on the left and which on the right. Since, for n observations there are $n - 1$ merges, there are 2^{n-1} possible orderings for the leaves in a cluster tree, or dendrogram. The algorithm used in hclust is to order the sub-tree so that the tighter cluster is on the left (the last, i.e., most recent, merge of the left sub-tree is at a lower value than the last merge of the right sub-tree). Single observations are the tightest clusters possible, and merges involving two observations place them in order by their observation sequence number.

An object of class hclust which describes the tree produced by the clustering process. The object is a list with components, which are summarized by the following table:

Table 3.2 The output components of the object resulting from the hclust() function and their description.

Value	Description
merge	An n-1 by 2 matrix. Row i of merge describes the merging of clusters at step i of the clustering. If an element j in the row is negative, then observation -j was merged at this stage. If j is positive then the merge was with the cluster formed at the (earlier) stage j of the algorithm. Thus negative entries in merge indicate agglomerations of singletons, and positive entries indicate agglomerations of non-singletons
height	A set of n-1 real values (non-decreasing for ultrametric trees). The clustering height: that is, the value of the criterion associated with the clustering method for the particular agglomeration
order	A vector giving the permutation of the original observations suitable for plotting, in the sense that a cluster plot using this ordering and matrix merge will not have crossings of the branches
labels	Labels for each of the objects being clustered
call	The call which produced the result
method	The cluster method that has been used
dist.method	The distance that has been used to create d (only returned if the distance object has a "method" attribute)

There are print, plot and identify methods and the rect.hclust() function for hclust objects. The plclust() function is basically the same as the plot method, plot.hclust, primarily for back compatibility with S-PLUS. Its extra arguments are not yet implemented.

Opinion clustering (Expectation-Maximization Algorithm):

The last step of our framework is the goal step of fitting each tweet into a cluster of possible opinions (vector of sentiment). EM assigns a probability distribution to each tweet which indicates the probability of it belonging to each of the clusters. EM can decide how many clusters to create by cross validation, or by previously specifying how many clusters to generate.

Generally, EM algorithm is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM algorithm is the most suitable clustering algorithm since it enables parameter estimation in probabilistic models with incomplete data. In our model the latent (hidden) variable here is the source of the opinion, where it could be a news channel or other external influences (i.e. other tweets, classmates, co-workers, friends, family, etc.). In order to simplify the explanation we start

with giving an example of a simple opinion tracking experiment. Lastly, we show how we used Weka to find the clusters' mean and standard deviation.

Consider a simple opinion tracking experiment in which we track the sentiment of two Twitter pages managed by two news channels with unknown biases θ_A and θ_B respectively (channel A has a positive sentiment towards a topic with probability θ_A and negative sentiment with probability $1 - \theta_A$ and similarly for channel B). Our goal is to estimate $\theta = (\theta_A, \theta_B)$ by repeating the following procedure five times: randomly choose one of the two channels, and perform ten independent sentiment assignments posted by the selected channel about a single topic. Thus, the entire procedure involves a total of 50 tweets analyzed (table 4.1).

Table 4.1 The complete case of the opinion tracking experiment.

Channel ID	**Sentiment of 10 tweets**	**Channel A's sentiment counts**	**Channel B's sentiment counts**
B	+, -, -, -, +, +, -, +, -, +		5 +, 5 -
A	+, +, +, +, -, +, +, +, +, +	9 +, 1 -	
A	+, -, +, +, +, +, +, -, +, +	8 +, 2 -	
B	+, -, +, -, -, -, +, +, -, -		4 +, 6 -
A	-, +, +, +, -, +, +, +,-, +	7 +, 3 -	
Total sentiment counts		24 +, 6 -	9 +, 11 -

During the experiment, suppose that we keep track of two vectors $x = (x_1, x_2, \dots, x_5)$ and $z = (z_1, z_2, \dots, z_5)$ where $x_i \in \{0,1, \dots,10\}$ are the number of

positive sentiment observed during the i_{th} set of tweets, and $z_i \in \{A, B\}$ is the identity of the channel used during the i_{th} set of tweets analyzed. Parameter estimation in this setting is known as the complete data case in that the values of all relevant variables in this model (the sentiment towards the topic and the news channel posted the set of tweets) are known. Here, a simple way to estimate θ_A and θ_B is to return the observed proportions of positive sentiment for each channel:

$$\hat{\theta}_A = \frac{\#\ of\ poistive\ sentiment\ posted\ by\ channel\ A}{total\ \#\ of\ tweets\ posted\ by\ channel\ A\ about\ the\ topic}$$

$$\hat{\theta}_B = \frac{\#\ of\ poistive\ sentiment\ posted\ by\ channel\ B}{total\ \#\ of\ tweets\ posted\ by\ channel\ B\ about\ the\ topic}$$

This intuitive guess is, in fact, known in the statistical literature as maximum likelihood estimation (the maximum likelihood method assesses the quality of a statistical model based on the probability it assigns to the observed data). If $logP(x, z; \theta)$ is the logarithm of the joint probability (or log-likelihood) of obtaining any particular vector of observed positive sentiment counts x and channel identities z, then the formulas above solve for the parameters $\hat{\theta} = (\hat{\theta}_A, \hat{\theta}_B)$ that maximize $logP(x, z; \theta)$.

Now consider a more challenging variant of the parameter estimation problem in which we are given the recorded positive sentiment counts x but not the identities z of the channels that posted each set of the tweets. We refer to z as hidden variables or latent

factors, which in our model represent the source of opinion which we want to reveal. Parameter estimation in this new setting is known as the incomplete data case. This time, computing proportions of positive sentiment for each channel is no longer possible, because in this setting we assume do not know the source of the tweet. However, if we had some way of completing the data (guessing correctly which channel posted in each set of the tweets), then we could reduce parameter estimation for this problem with incomplete data to maximum likelihood estimation with complete data.

One iterative scheme for obtaining completions could work as follows: starting from some initial parameters, $\hat{\theta}^{(t)} = (\hat{\theta}_A^{(t)}, \hat{\theta}_B^{(t)})$ determine for each of the five sets whether channel A or channel B was more likely to have posted the observed tweets (using the current parameter estimates). Then, assume these completions (guessed channel assignments) to be correct, and apply the regular maximum likelihood estimation procedure to get $\hat{\theta}^{(t+1)}$. Finally, repeat these two steps until convergence. As the estimated model improves, so too will the quality of the resulting completions.

The expectation maximization algorithm is a refinement on this basic idea. Rather than picking the single most likely completion of the missing channel assignments on each iteration, the expectation maximization algorithm computes probabilities for each possible completion of the missing data, using the current parameters $\hat{\theta}^{(t)}$. These probabilities are used to create a weighted training set consisting of all possible completions of the data. A modified version of maximum likelihood estimation that deals

with weighted training examples provides new parameter estimates, $\hat{\theta}^{(t+1)}$. By using weighted training examples rather than choosing the single best completion, the expectation maximization algorithm accounts for the confidence of the model in each completion of the data (fig 4).

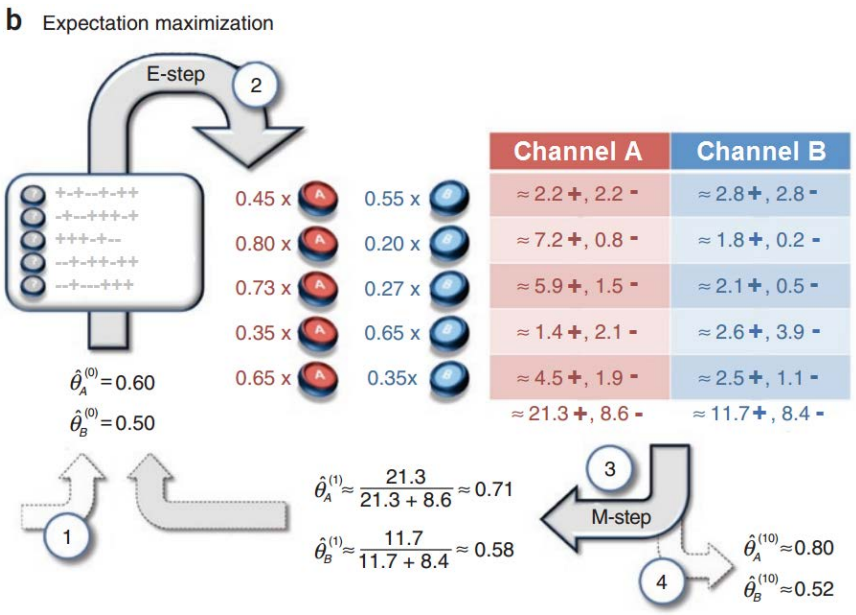

Figure 4 The incomplete case of the opinion tracking experiment.

However, the implementation of this model is not the exact incomplete case we are aiming for, thus we modify conditions of the previous incomplete case experiment as following. First, the sets of ten sentiment values are driven from the same tweet about multiple different 10 topics, which are defined apriori and constant among all tweets. Second, the probabilities computed in the expectation step according to the distributions are for all sets vertically in figure 4, since each set is now considered as one tweet. Third,

the tweets analyzed are from anonymous users affected by multiple opinion sources. It is important to point that we are not concerned about the identity of the user; we are concerned about the source of the opinion which the sentiment is based upon. Lastly, the sentiment values used do not necessarily have to be trivial polarity (positive and negative); they could be sentiment groups or scores. Thus, in our model, the aim of the expectation step is not to find a single value for $\hat{\theta}$, but is to fit a normal distribution onto the sentiment observed among the tweets. This means that the EM algorithm initially assumes all of the analyzed tweets are in one cluster with a normal distribution. Then the algorithm applies the maximum likelihood procedure to improve the assumed parameters, which could result into splitting the guessed cluster into two, and so on.

The expectation maximization algorithm alternates between the steps of guessing a probability distribution over completions of missing data given the current model (known as the E-step) and then re-estimating the model parameters using these completions (known as the M-step). The name ‘E-step’ comes from the fact that one does not usually need to form the probability distribution over completions explicitly, but rather need only compute ‘expected’ sufficient statistics over these completions. Similarly, the name ‘M-step’ comes from the fact that model re-estimation can be thought of as ‘maximization’ of the expected log-likelihood of the data. Introduced as early as 1955 by Ceppellini et al. in the context of gene frequency estimation, the expectation maximization algorithm was analyzed more generally by Hartley and by Baum et al. in

the context of hidden Markov models, where it is commonly known as the Baum-Welch algorithm. The standard reference on the expectation maximization algorithm and its convergence is Dempster et al in 1977.

EM using Weka:

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from a Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. The software is available for free online, and installed on our fedora server at CAU. The command line to start Weka is:

java -jar /opt/weka-3-6-9/weka.jar &

The first window that appears is the Weka's graphical user interface (GUI) chooser as shown below:

Figure 5 Weka's GUI Chooser

The Weka explorer window is easy to use for importing data in ARFF format and applies several machine learning algorithms. The Header of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types followed by the data. Also, we show figure 6 as an example of an imported dataset, through the ‘Open file’ button:

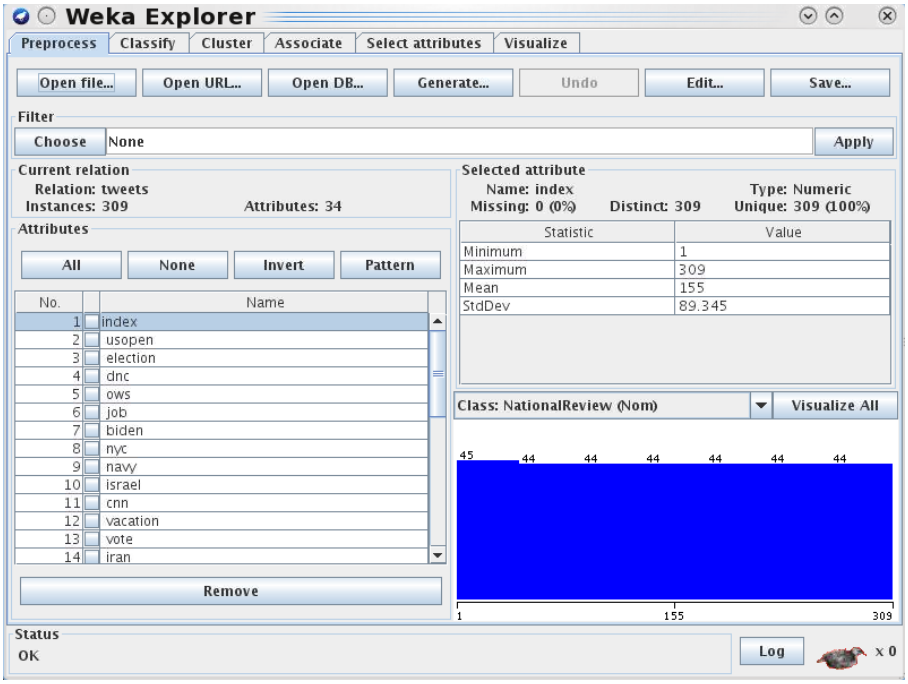

Figure 6 Weka’s GUI Explorer

The “Cluster” tab gives several options of clustering algorithms (i.e. Cobweb, DBScan, FarthestFirst, FilteredClusterer, etc.). However, we are concerned with using the EM algorithm, which can be chosen through the ‘Choose’ button as shown in the figure below:

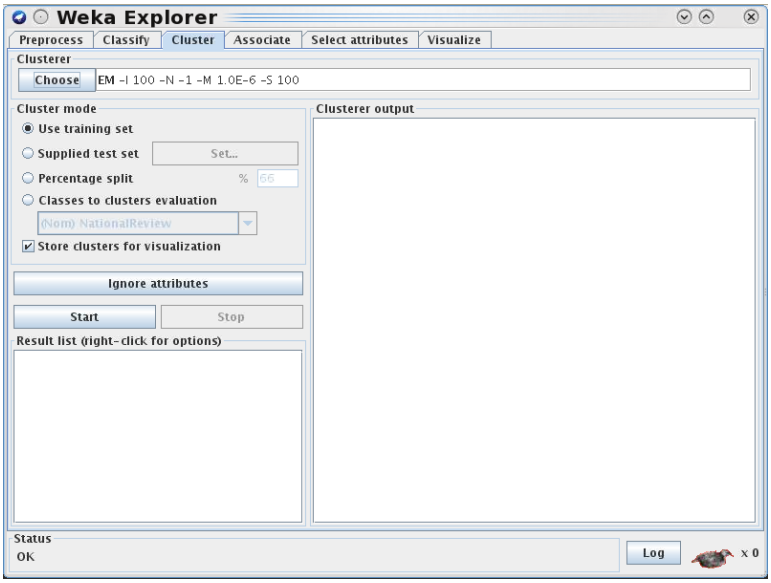

Figure 7 Weka's Cluster tab

Along with assigning the sentiment of each tweet we search for keywords relative to the news channels analyzed in the tweet. Thus, with each tweet we have information about which news channel is the tweet referring to. As discussed in the hypothesis section, it is very important to calculate the percentage of referrers in herds of opinions. This type of information is assigned as two nominal values $\{news, Nonews\}$. For example, if the tweet contains $news$ at the fox news column, but has $Nonews$ at the CNN news column, then this tweet has referred its opinion from fox news but not from CNN. In this step we ignore the news refers, using the ignore attributes button, values for all news channels addressed, because we do not want these attributes play a role in the clustering algorithm. We only need these attributes to show us on the visual an approximate analysis of the percentage of referrals in diverse and herds of opinions.

In Weka, the clustering scheme generates probabilistic descriptions of the clusters in terms of mean and standard deviation for the numeric attributes and value counts (incremented by 1 and modified with a small value to avoid zero probabilities) - for the nominal ones. We investigate the mean and the standard deviation of each cluster in order to find the overlapping and the isolated clusters. In "Classes to clusters" evaluation mode this algorithm also outputs the log-likelihood, assigns classes to the clusters and prints the confusion matrix and the error rate.

EM assigns a probability distribution to each instance which indicates the probability of it belonging to each of the clusters. EM can decide how many clusters to create by cross validation, or you may specify apriori how many clusters to generate.

The cross validation performed to determine the number of clusters is done in the following steps:

1. The number of clusters is set to 1
2. The training set is split randomly into 10 folds
3. EM is performed 10 times using the 10 folds
4. The log likelihood is averaged over all 10 results
5. If log likelihood has increased the number of clusters is increased by 1 and the program continues at step 2

The number of folds is fixed to 10, as long as the number of instances in the training set is not smaller than 10. If this is the case the number of folds is set equal to the number of instances[7].

[7] http://www.cs.waikato.ac.nz/ml/weka/

CHAPTER 4

RESULTS & DISCUSSION

Data Collection:

Under the administration of Professor Peter Molnar over 170 million tweets were harvested using a stream that was active since September 2012 to monitor the current political situation around the world. The Twitter project was established on the fedora server to grant the access to this database to the faculty and the students of CAU, and researchers affiliated with the institution. The website hosts all detailed information at the fedora website at[1].

We chose the 140dev streaming API to store the tweets into our fedora using MySQL database. The 140dev API framework is a free source code library written by Adam Green[2] and released under the General Public License (GPL). The goal of this API is to provide a simple interface to the Twitter Streaming API. The current version provides a tweet aggregation database, and a plugin for tweet display on any Web page. However, Mr. Green is planning to provide plugins for data mining, automated tweeting and account management in the future. 140dev is written in PHP and JavaScript, and uses the MySQL database for storage. Thus all our extraction queries that we present in the thesis are in MySQL. All of the interactions between the modules in this framework are

[1] http://fedora.cis.cau.edu/~pmolnar/TWITTER/
[2] http://140dev.com/

through the database, which means that additional modules can be written in any language that has a MySQL interface of 140dev. Additionally, for developers' interests, the API provides flexibility in expanding, which is one of the reasons for calling it a framework. The framework is composed of the database server and other plugins. The database server is the core module of the 140dev API. It uses the Twitter API to gather tweets for selected keywords and stores them in a MySQL database. The rest of the libraries are built as plugins that share information with this database server. One of the important plugins that most advertising websites used to add Twitter widgets is the display plugin. The plugin calls the copy of the Twitter database server, retrieves the most recent tweets, and returns them as formatted HTML. All tweet entities are rendered as links.

In order to monitor the political situation with respect to coverage of mainstream media, we chose particular terms to be used in streaming the tweets.

Database Structure:

In our relational database we have 10 tables connected together, which contains information about the users, tweets, tweet URLs, tags and mentions, mentions' counts, JSON cache, the degrees and their in and out. Figure 8 clarifies the fields in each table with their type, whether they are NULL or not, their keys, their default values and an extra comment. In the figure below we reveal the relational database tables' fields and how they are connected.

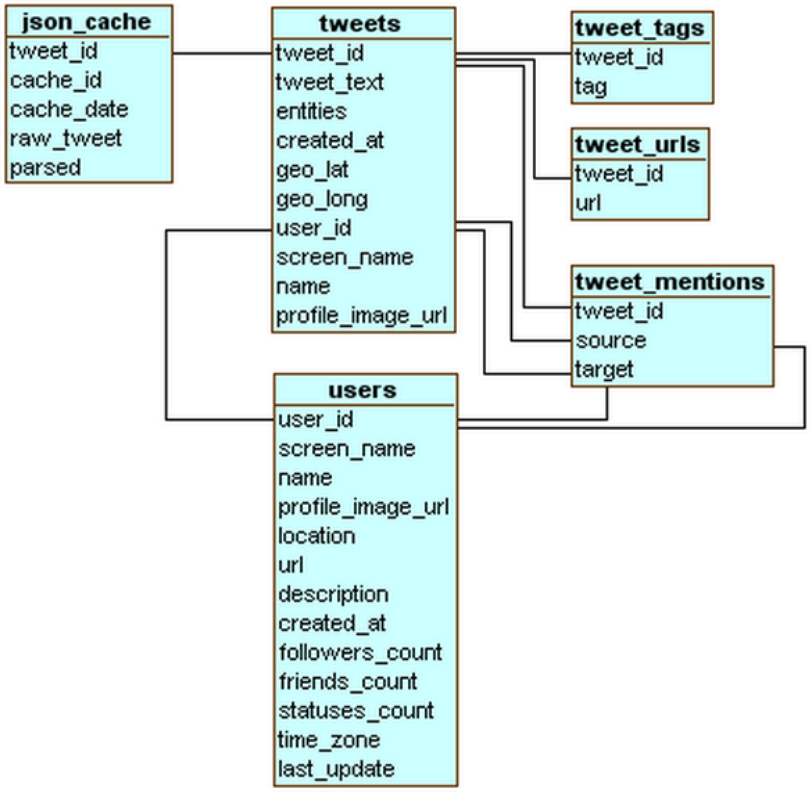

Figure 8 The relational database table's fields and their connections.

Statistical analysis:

Our statistical analysis on the percentage of mainstream media mentions among the total number of tweets was conducted for 10 million tweets. The following table shows news channels' names, keywords used for their search and their frequency:

Table 5.1 The number of tweets which mentioned the following news channels and the used keywords to search for them.

Channel's name	Search keywords	Counts
CNN	#cnn	24,354
ABC news	#abc/@abc/abc news/abcnews	23,100
Reuters	reuters	22,896
NBC news	#nbc	18,426
Fox News	Foxnews/fox news	16,798
BBC	bbc	11,198
Associated Press	@ap/#apassociated press/associatedpress	10,963
NY Times	Nytimes/nytimes/newyorktimes/ny times/new york times	8,351
Washington Post	washington post/washingtonpost	6,178
USA Today	usa today/usatoday	7,879
Agence France-Presse	agence france presse/ agencefrancepress /afp	3,076
Forbes	forbes	2,981
bloomberg	bloomberg	1,981
Wall Street Journal	wallstreetjournal/wallstreet journal	1,484
TMZ	tmz	1,134
Total		149,073 = 1.49%

Some search keywords mislead the counts of mentions as they might be simple components in normal words, for example, "ap" and "abc". By just using "ap" to count the frequency of AP news mentions among the 10 million tweets the result was 475,951

tweets. However, normal words like “apple” or “appeal” contain the “ap” keyword, which means it that some tweets were false counted by only considering this simple combination of letters. Thus, we had to restrict the counts by combining with “#” and “@”.

While table 5.2 shows the percentage of original tweets (not RT) versus the number of original that have links. This study helped us investigate the significance of sharing links among the users, which could be a door for another type of research question in the future work, for example, analyzing the links’ web pages or documents to enhance the sentiment analysis of the tweet.

Table 5.2 The percentage of original tweets and the original ones that have links

Number of analyzed tweets	**Original**	**Original & has link**
10,000,000	5,881,697 (58.8%)	2,719,402 (27.2%)

Table 5.3 shows the percentages of tweets which have one adjective and more than one adjective in the same tweet out of 100,000 tweets.

Table 5.3 The percentages of tweets which have one adjective and more than one adjective

Number of analyzed tweets	**One adjective**	**More than one adjective**

100,000	13,668 (13.7%)	6,103(6.1%)

We show the analysis settings of our experiments that produce the clusters which express the sources of opinions as hidden variables. We used different inputs and filtering categories for the finding the trending topics and the sentiment assignment steps, as was shown in figure 1 in the previous chapter. In this section we chow the categories of filtered used and the combination of different analysis settings.

In the framework we apply different types of filtering categories. The filtering category depends on the property on which the tweets are filtered upon. The figure below shows the possible filtering properties that can be applied on each line:

In the table below, we summaries the category versus the property of filtering and the definition of property:

Table 6.1 Definitions of the filtering categories.

Property	**Definition & Reasoning**
RT	RTs are not the scope of our analysis, and considered as noisy data
News	Tweets which have at least one news channel mentioned
1-Topic	Tweets which have at least one topic mentioned
n-Topic	Tweets which have at least n topics mentioned
Adjective	Tweets which have only one adjective describing its sentiment

We filter out the RTs, unlike Myers Seth et al., since our scope is focused on finding the influence through comparing the sentiment of original tweets. Basically, it is worthless to analyze opinions which contain all zero vector, and that could result from either no adjective used or a trending topic mentioned. And thus finding the frequent itemsets plays its role in reducing matrix sparsity, so when the sentiment is assigned to a topic we guarantee with high probability that the tweet would contain another topic. This is also the same reason, we use the adjective filter to decrease the sparsity of the sentiment matrix used in the opinion clustering step. Nevertheless, we exclude the tweets which have more than one adjective, since we cannot handle multi-sentiment tweets. We do not apply any technique to differentiate the reference of each adjective in a multi-sentiment tweet. We also apply the one-topic filter to guarantee at least one topic mentioned per analyzed tweet, thus it is mandatory. However, it is not necessary to filter using n-topic filters.

Trending Topics:

In the trending topics step we apply the Apriori algorithm on two different groups of words: the most frequent general (not hashtags specifically) 30 words and on all hashtags. When collecting the tweets for both settings we filter the RTs out. However, the difference in application is due to the purpose of using the outputs of both settings. When we use all hashtags we are looking at the most frequent itemsets to be the trending topics.

While when using the most general 30 words we look for the association rules between those 30 words and the news channels. The purpose is to use associated words to the news channels, in the future, to conduct validation analysis using the web archives of the news channels. The articles searched by the general frequent words would be compared with the tweets sentiment wise. We avoid using the hashtags since they are very particular to the tweeting behavior of the users, and many hash-tagged words are not usable for searching news archives.

Using hashtags:

In the script shown in appendix A.2, we start with harvesting and filtering the tweets by RT category, then sort the hashtags to come up with counts shown in the previous list. The 'candHash.py' functionality is to construct the transaction matrix, as shown in appendix A.3. As the Apriori algorithm's implementation was shown in the last chapter, it uses numbers to index the hashtags for simplifying the input for the program, especially, because it is written in C++, appendix A.1. Lastly, we map the resulted indexes into the actual hashtags, by the 'num2Hash.py' program as shown in appendix A.4. The minimum support count was adjusted according to the average of the counts of all hashtags. A very low minimum support count would result into a computationally expensive implementation and consider low frequent unimportant topics. While choosing a high support count would result into few hashtags and ignore important topics. Thus,

we considered the average of all 1-frequent itemsets to be the minimum support count, which is 5,246.

The last frequent itemset contains 8 items and figure 9 shows the counts of the top 10 frequent itemsets. Each hashtag in the frequent itemsets is a topic. We ended up with the following list of 30 topics:

[obama, usa, tcot (Top Conservatives On Twitter), p2 (Progressive Propaganda), news, cnn, romney, teaparty, tiot (Top Independents On Twitter), usopen, dnc (Democratic National Committee), teamfollowback or tfb (you will follow back), economy, election, iran, israel, job, media, navy, nyc (New York City), ows (Occupy Wall Street), politics, twisters, usopen (Tennis Championship), vote, jakarta, london, politics, republican]

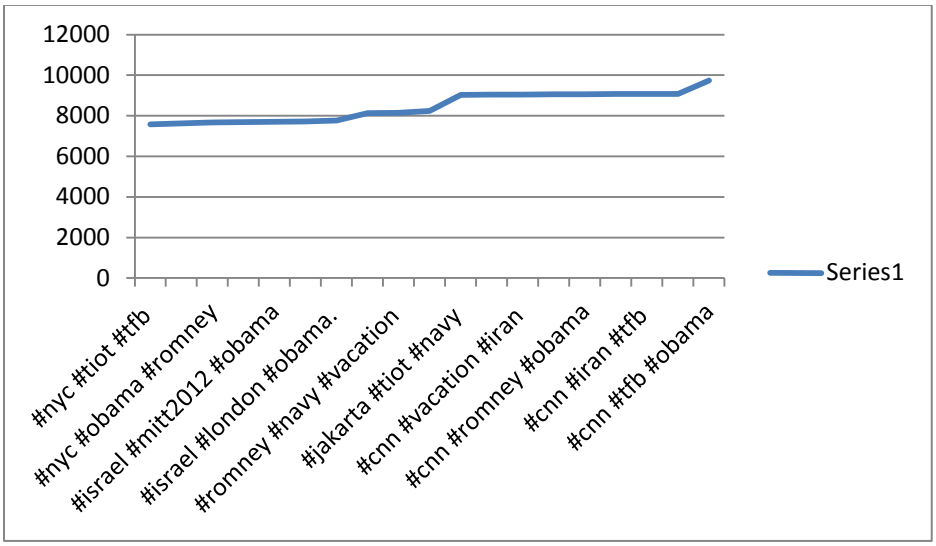

Figure 9 The frequencies of the frequent itemsets.

The short hashtags which have political meaning are considered, and thus, we do not use hashtags to find association rules between topics and channels, since such short words could be misleading for the search engines. There are repeated entities being expressed by different hashtags like “mittromney” and “mitt”. We combined those hashtags as the same by part of word searching, and the matches are recognized as the same entity. Thus, in the sentiment assignment step we use all possible hashtags that are used to express the same topic or entity. We used the website [3]to define the short hashtags.

Association rules:

As mentioned in the head of this section, we are searching for association rules between general 30 frequent words and the news channels to be used in searching articles in the news web archives, where these articles in our future work will be compared with the sentiment of the tweets, as a validation schema. We found that 20 is the average count of 1-frequent itemsets, which lead to gaining at least 2 search keyword per channel. The following table summarizes those keywords and shows the calculation of their confidence.

Table 6.2 Association rules between the channels and the most 30 frequent words and their confidence level.

[3] http://tagdef.com/

Frequent itemsets	Support count with channel	Support count without channel	Confidence level
NBC:			
Romney, Obama	140	306	45.8%
Romney, Health	30	144	20.8%
Obama, Health	70	280	25%
Obama, Job	70	110	63.6%
NY Times:			
Romney, Job	100	533	5.3%
Romney, Taxes, Republican	20	63	31.7%
Romney, Gas, Employment	55	140	39.3%
Reuters:			
Obama, Mitt	497	514	96.7%
Romney, Obama, Job	222	306	72.5%
Fox:			
Romney, Elections	220	650	33.8%
Obama, Health	240	280	85%
ABC:			
Obama, Romney	30	306	9.8%
Romney, Economy	25	84	29.8%
CNN:			
Obama, Employment	55	70	78.6%

Romney, Taxes	20	63	31.7%

The higher the confidence level of keywords that appear with a channel the more it is suitable to be used for searching in the web archive to find related articles from that particular channel. The percentages marked in red are the frequent itemsets chosen to be associated with the channels marked.

Observations & Inferences:

In this section we show our observations and the inferred meanings from the opinion clustering step through graphs and statistics calculated for each experiment setting using Weka. The original results from the scoring sentiment assignment method are shown first, and then we compare these results using the adjective hierarchy sentiment assignment method.

As was shown in the methodology chapter, Weka Explorer provides a GUI to load data, preprocess it, and then apply various types of machine learning algorithms on the data. The Weka Explorer also provides the option of ignoring attributes and choosing the adequate evaluation settings. By using the “training set”, this is the default evaluation choice, Weka classifies the training instances into clusters according to the cluster representation and computes the percentage of instances falling in each cluster after

generating them. For probabilistic cluster representation, it is more suitable to evaluate clustering on a separate test dataset using “Supplied test set”. This option provides loading a file or linking to a web page. The third and last method of evaluation in Weka is by assigning classes to clusters based on the majority value of the class attribute within each cluster. Then Weka computes the classification error, based on this assignment and also shows the corresponding confusion matrix. This option is done by choosing “Classes to clusters evaluation”. Nevertheless, we use the default “training set” option, since we do not have separate test set available.

Experiment 1:

Here we show our observations of the opinion clustering step when using the scores list. The setting of the experiment is shown in figure 15 to view the applied filtering categories through the framework. We used the fine-grained filtering of 3-topics per tweet to restrict the sparsity of the matrix and obtained more valuable results.

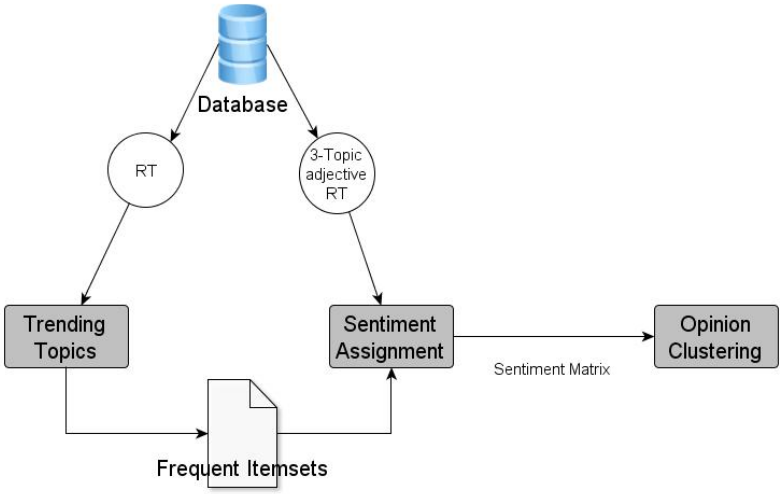

Figure 10 The category of filters applied through the framework for experiment 1.

The resulting overall clustered instances are distributed as shown in table 7.1, where 10 clusters were selected. There are no inferences that could be derived from that table; it just shows the distribution of instances among different clusters.

Table 7.1 The percentages of distributions of clusters for experiment 1.

Cluster number	**Number of instances (Percentage)**
0	306 (8%)
1	93 (2%)
2	43 (1%)
3	17 (0%)
4	42 (1%)
5	973 (26%)
6	1043 (28%)

7	639 (17%)
8	517 (14%)
9	73 (2%)

We present the distribution of the sentiment towards each topic among the clusters using the mean and standard deviation in table 7.2.In this table we only shows the minimum and maximum of all clusters for topics which has isolated clusters, and mark those isolated clusters in red.

Table 7.2 The distribution of clusters among the sentiment towards each topic using the mean and the standard deviation.

Topic	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8	Cluster 9
vote										
mean	0	1.0079	0.2488	0	0	0.2105	0.0183	0.2148	-2.5308	0.0177
std. dev.	1.151	1.2403	1.2477	1.151	1.151	0.5628	0.1339	1.183	1.1214	0.1668
Min	-+0.5755	0.38775	-0.37505	-+0.5755	-+0.5755	-0.0709	-0.04865	-0.3767	-3.0915	-0.0657
Max		1.62805	0.87265			0.4919	0.08525	0.8063	-1.9705	0.104
Iran										
mean	0.026	0	0	0	-0.2387	0	1.0247	0	0	0
std. dev.	0.2537	0.228	0.228	0.228	0.7768	0.0001	1.47	0.228	0.228	0.228
Min	-0.10085	-+0.114	-+0.114	-+0.114	-0.6271	-	0.2897	-+0.114	-+0.114	-+0.114
Max	0.15285				0.1497	+0.00005	1.7597			
Romney										
mean	0.0768	0.0295	0.5792	-1.3168	0.0299	2.1851	-0.0684	0.3018	-0.5452	0
std. dev.	0.3878	0.2082	1.1817	1.2391	0.2428	0.89	0.4479	1.0174	1.3661	1.3364
Min	-0.1171	-0.0746	-0.01165	-1.93635	-0.0915	1.7401	-0.29235	-0.2069	-1.2282	-+0.6682
Max	0.2707	0.1336	1.17005	-0.69725	0.1513	2.6301	0.15556	0.8105	0.13785	
Obama										

Mean	1.6478	1.9145	1.274	-1.6177	-2.0128	1.6149	1.501	0.5893	-2.5957	2.8498
std. dev.	0.7483	1.2932	1.7535	1.2721	1.771	1.3668	1.4776	1.9736	1.1649	0.5306
Min	1.27365	1.2679	0.39725	-2.25375	-2.8983	0.9315	0.7622	-0.3975	-3.17815	2.5845
Max	2.02195	2.5611	2.15075	-0.98165	-1.1273	2.2983	2.2398	1.5761	-2.01325	3.1151

From this table we can observe that cluster number 8 is an isolated cluster with respect to the "vote" topic, where the range of sentiment used by this cluster is between -1.9701 and -3.0915. The rest of the clusters express their sentiment out of this range. While for the topic "Iran" we can see that cluster number 6 is isolated from the rest at the range between 0.2897 and 1.7597. The same for topics "Romney" and "Obama", the clusters which exhibited isolation by not overlapping with other clusters, their minimums and maximums are marked in red.

According to the table, in this sense it is obvious in the figures 16-19 that topics "vote", "Iran", "Mitt" and "Obama" have different segregated *unidimensional* opinions. These figures show the clusters versus the sentiment towards each of the mentioned topics. Weka's visualizing tool show the segregation using the jitter option, which is quite unclear. Thus, for clearer image about the isolated clusters figures 20-23 show simple area graph plots of the minimum and maximum for topics that show isolated clusters.

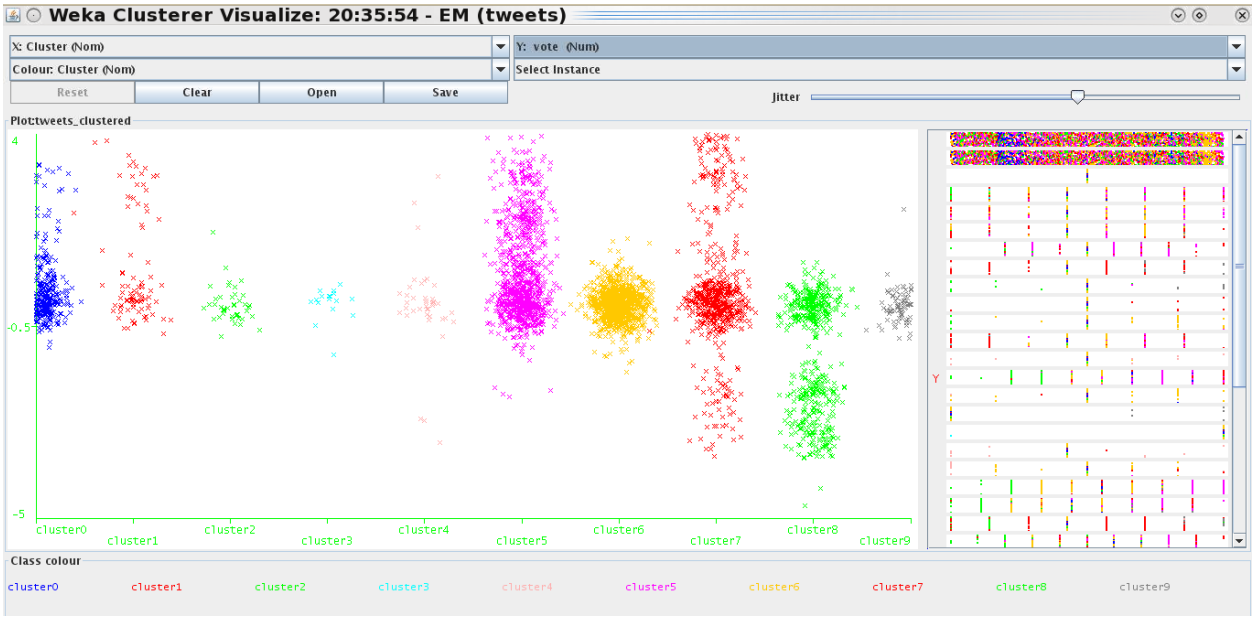

Figure 11 The distribution of clusters among the sentiment towards the topic “vote”, where cluster 8 is the isolated cluster on Weka’s visual.

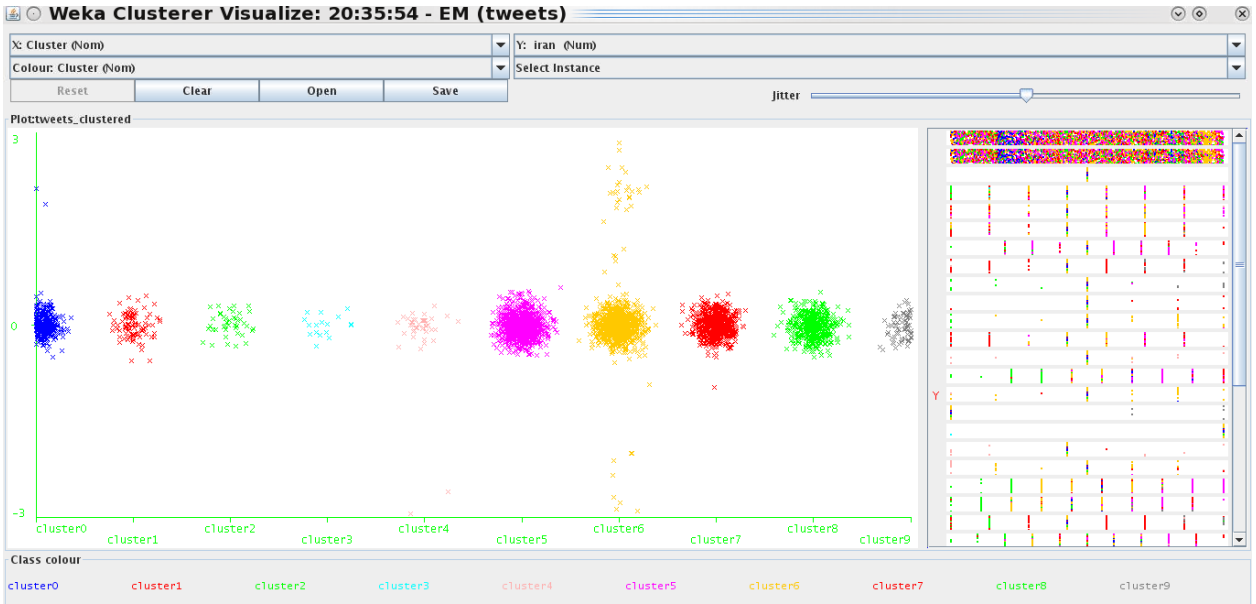

Figure 12 The distribution of clusters among the sentiment towards the topic “Iran”, where cluster 6 is the isolated cluster on Weka’s visual.

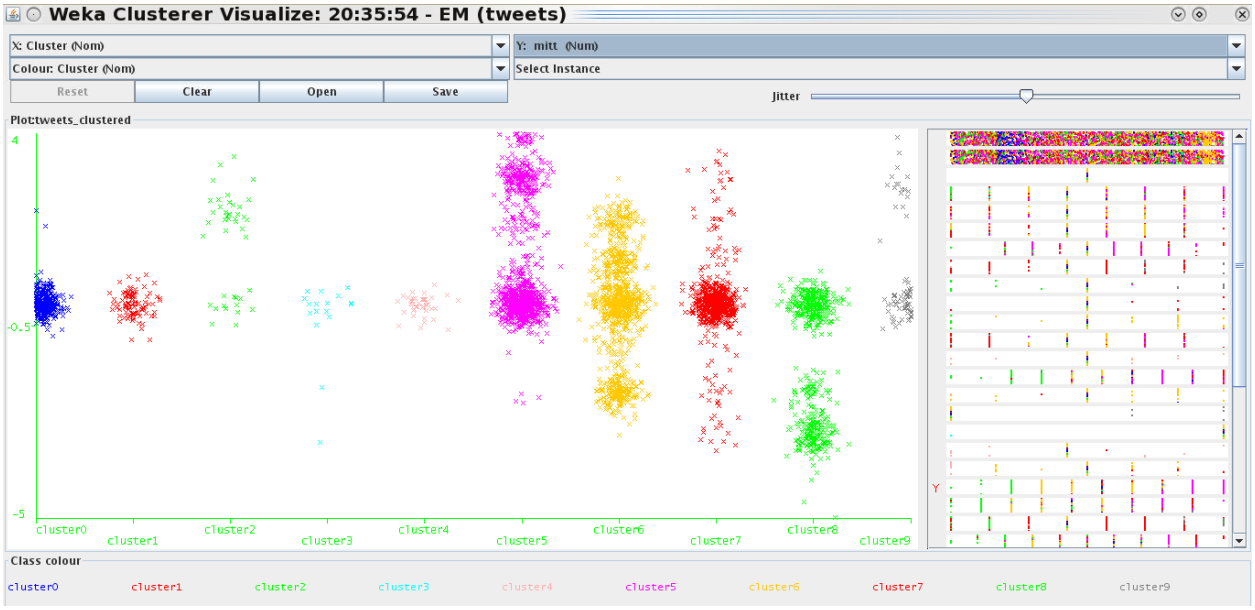

Figure 13 The distribution of clusters among the sentiment towards the topic “Mitt”, where cluster 5 is the isolated cluster on Weka’s visual.

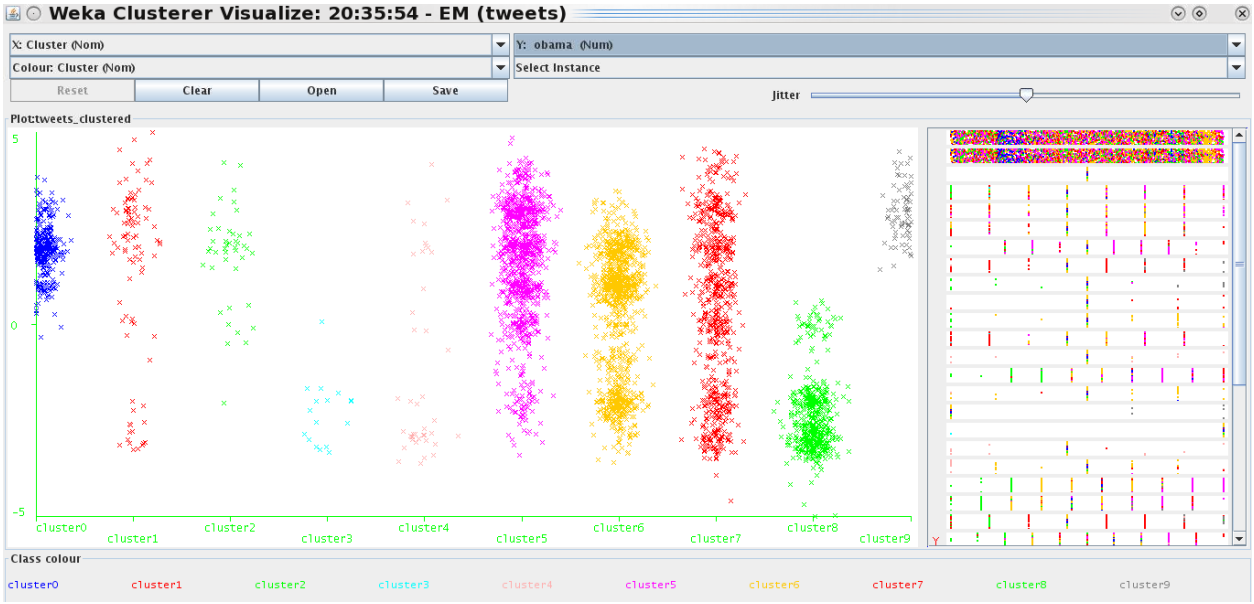

Figure 14 The distribution of clusters among the sentiment towards the topic “Obama”, where cluster 9 is the isolated cluster on Weka’s visual.

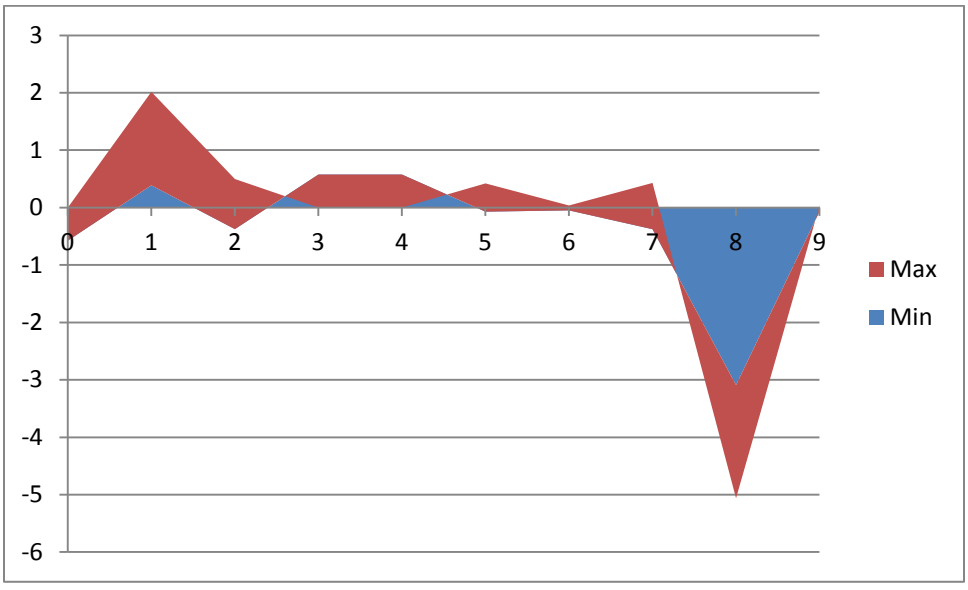

Figure 15 The distribution of clusters among the sentiment towards the topic “vote”, where cluster 8 is the isolated cluster on an area plot using the minimum and maximum values.

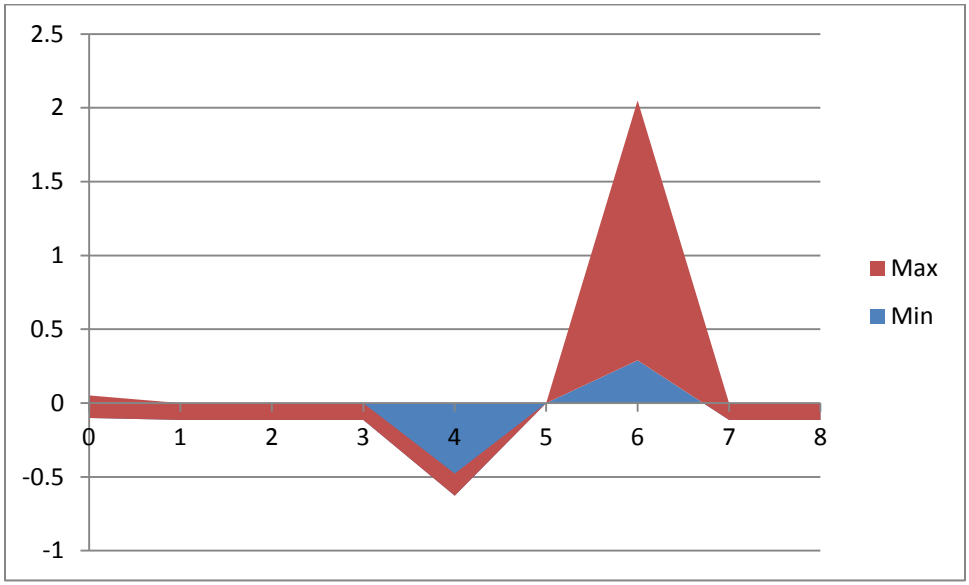

Figure 16 The distribution of clusters among the sentiment towards the topic “Iran”, where cluster 6 is the isolated cluster on an area plot using the minimum and maximum values.

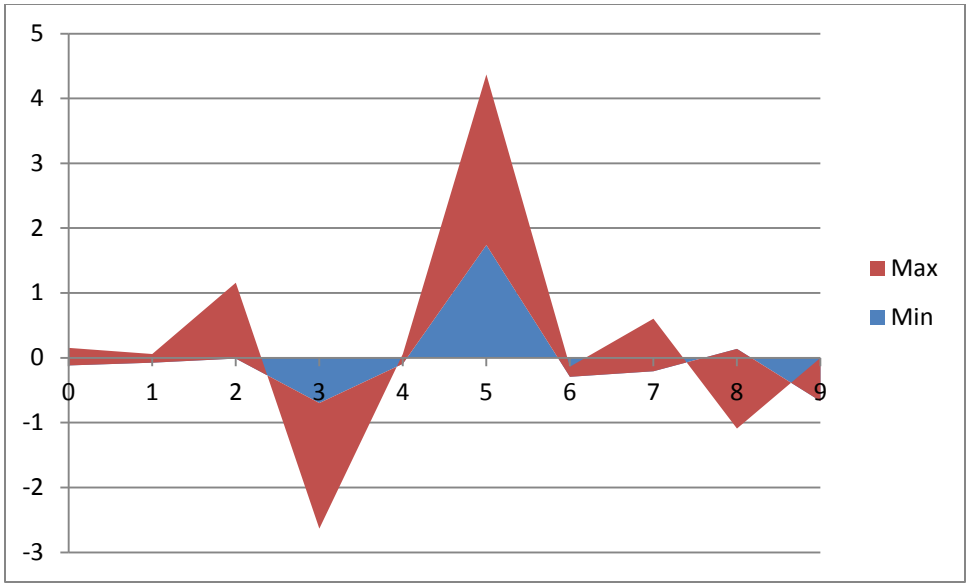

Figure 17 The distribution of clusters among the sentiment towards the topic “Mitt”, where cluster 5 is the isolated cluster on an area plot using the minimum and maximum values.

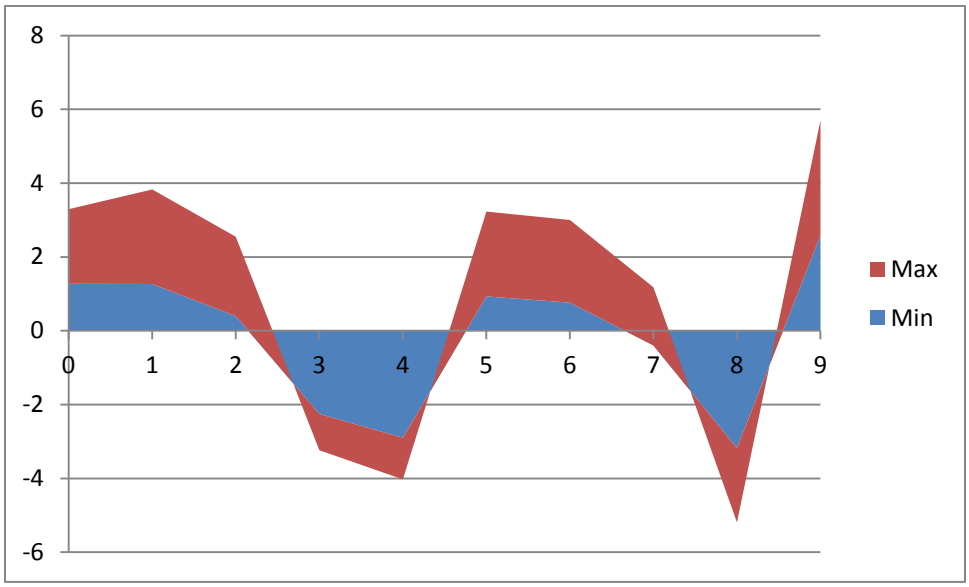

Figure 18 The distribution of clusters among the sentiment towards the topic “Obama”, where cluster 9 is the isolated cluster on an area plot using the minimum and maximum values.

Table 7.3 shows the number of times each news channel was referred in the isolated clusters. The numbers are significantly low, and thus we apply the news filter to focus our analysis on the news channels only in the next subsection.

Table 7.3 The number of mentions for each channel in topics with isolated clusters.

Topic>>Sentiment	ABC	NY times	Fox	CNN	Reuters	NBC	Total
Vote < -1.9705							
General	1	0	1	5	0	1	3308
Cluster Specific(8)	1	0	0	0	0	1	3308
Iran > 0.2897							
General	1	0	1	5	0	1	5
Cluster Specific(6)	1	0	0	0	0	1	1
Mitt > 1.7401							
General	1	0	1	10	0	3	2914
Cluster Specific(5)	1	0	0	3	0	2	2914
Obama > 2.5845							
General	4	3	6	46	0	12	3342
Cluster Specific(9)	2	0	2	13	0	4	2648

Experiment 2:

Another setting, we filtered out tweets which have no news reference at all. However, in order to increase the number of tweets analyzed, we made the topic filter set

at one only instead of three, as shown in figure 24. This setting has left for us 309 tweets only to be analyzed.

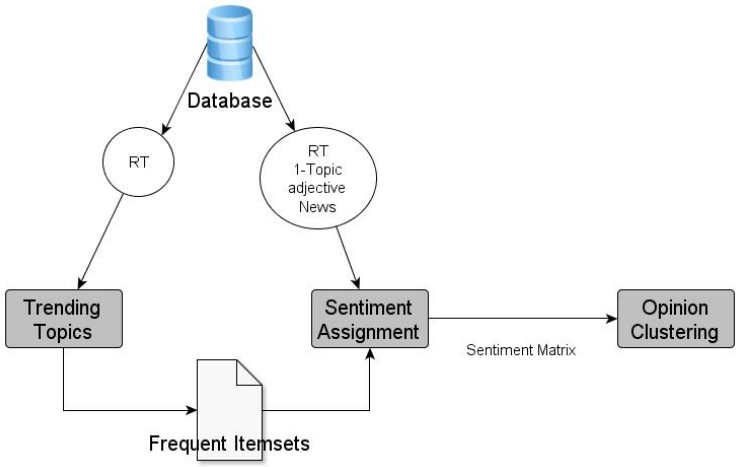

Figure 19 The category of filters applied through the framework for experiment 2.

This setting has resulted in 5 clusters selected, which took Weka 10.43 seconds. Table 8.1 shows the distribution of the instances among the clusters.

Table 8.1 The percentages of distributions of clusters for experiment 2.

Cluster number	**Number of instances (percentage)**
0	56 (18%)
1	70 (23%)
2	75 (24%)
3	81 (26%)
4	27 (9%)

We present the distribution of the sentiment towards each topic among the clusters using the mean and standard deviation in table 8.2. As we also red mark the clusters which express segregation from other clusters. In this table we only show the minimum and maximum of all clusters for topics which has isolated clusters, and mark those isolated clusters in red.

Table 8.2 The distribution of clusters among the sentiment towards each topic using the mean and the standard deviation for experiment 2.

Topic	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
OWS					
mean	0.056	1.312	0	0.3321	-0.1141
std. dev.	0.33	0.9767	0.5596	0.471	0.5327
Min	-0.109	0.82365	-0.2798	0.0966	-0.38045
Max	0.221	1.80035	0.2798	0.5676	0.15225
romney					
Mean	1.0096	0.8391	0.7513	0.8878	-1.1781
std. dev.	1.292	1.036	1.6515	0.3971	1.2884
Min	0.3636	0.3211	-0.07445	0.68925	-1.8223
Max	1.6556	1.3571	1.57705	1.08635	-0.5339
obama					
Mean	1.9783	1.8987	0.8593	0.9996	-2.4111
std. dev.	1.2179	0.4863	1.8018	0.242	0.6909
Min	1.36935	1.65555	-0.0416	0.8786	-2.75655
Max	2.58725	2.14185	1.7602	1.1206	-2.06565

From this table we can observe that cluster number 1 expresses segregation in opinion towards the Occupy Wall Street (OWS), where the range of sentiment used by this cluster is between 0.82365 and 1.80035. Cluster number 4 expressed segregation towards the topic “Romney”, where the range of sentiment used by this cluster is between -1.8223 and -0.5339, which is not very far from other ranges of sentiment used by other clusters. Lastly, clusters number 0, 1 and 4 express interesting isolation in opinion. Cluster 0 and 1 are isolated together in the positive region between 1.36935 and 2.58725 and cluster 4 is isolated in the negative region between -2.75655 and -2.06565.

According to the table, in this sense it is obvious in the figures 25-27 that topics “OWS”, “Romney” and “Obama” have different segregated *unidimensional* opinions. These figures show the clusters versus the sentiment towards each of the mentioned topics. For clearer image about the isolated clusters figures 28-30 show simple area graph plots of the minimum and maximum for topics that show isolated clusters.

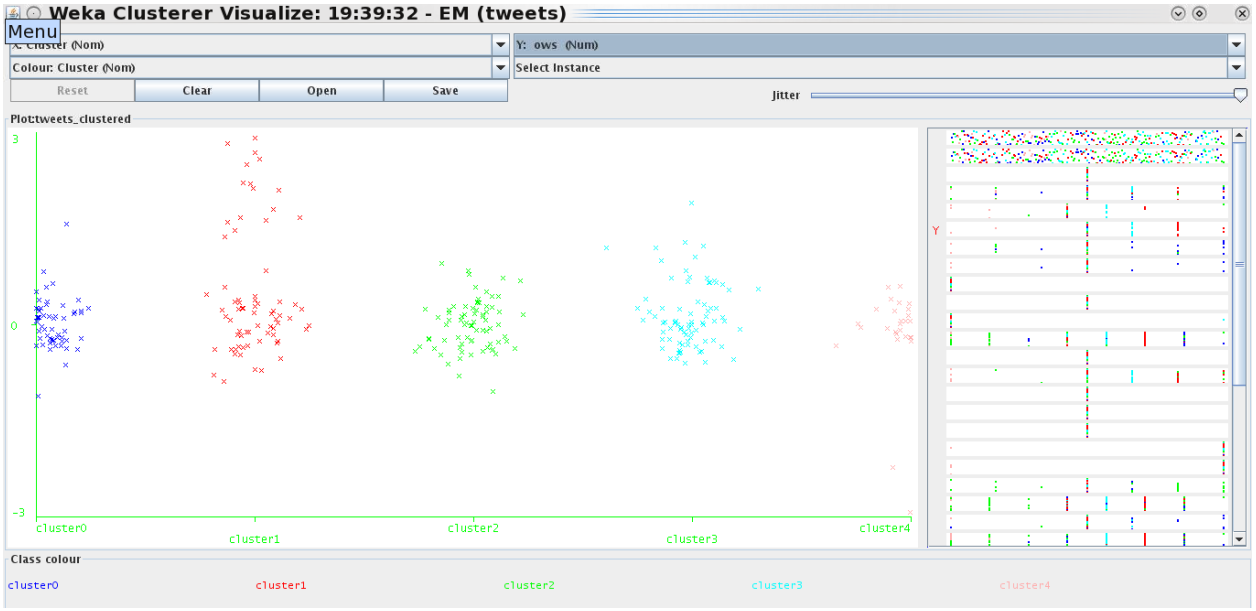

Figure 20 The distribution of clusters among the sentiment towards the topic “OWS”, where cluster 1 is the isolated cluster on Weka’s visual.

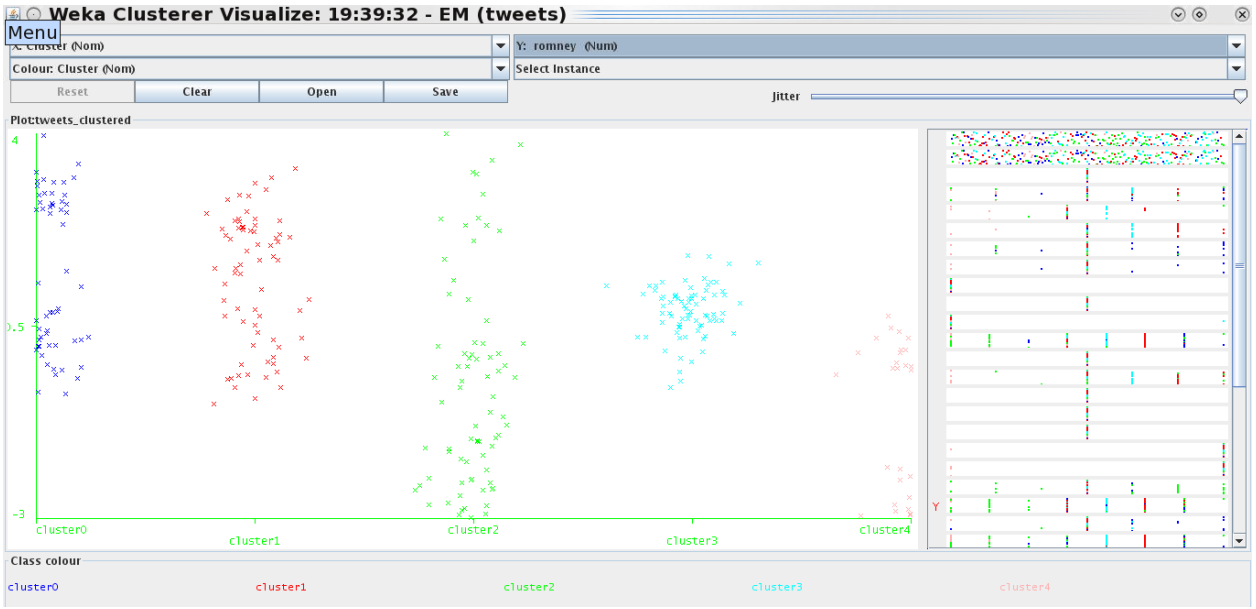

Figure 21 The distribution of clusters among the sentiment towards the topic “Romney”, where cluster 4 is the isolated cluster on Weka’s visual.

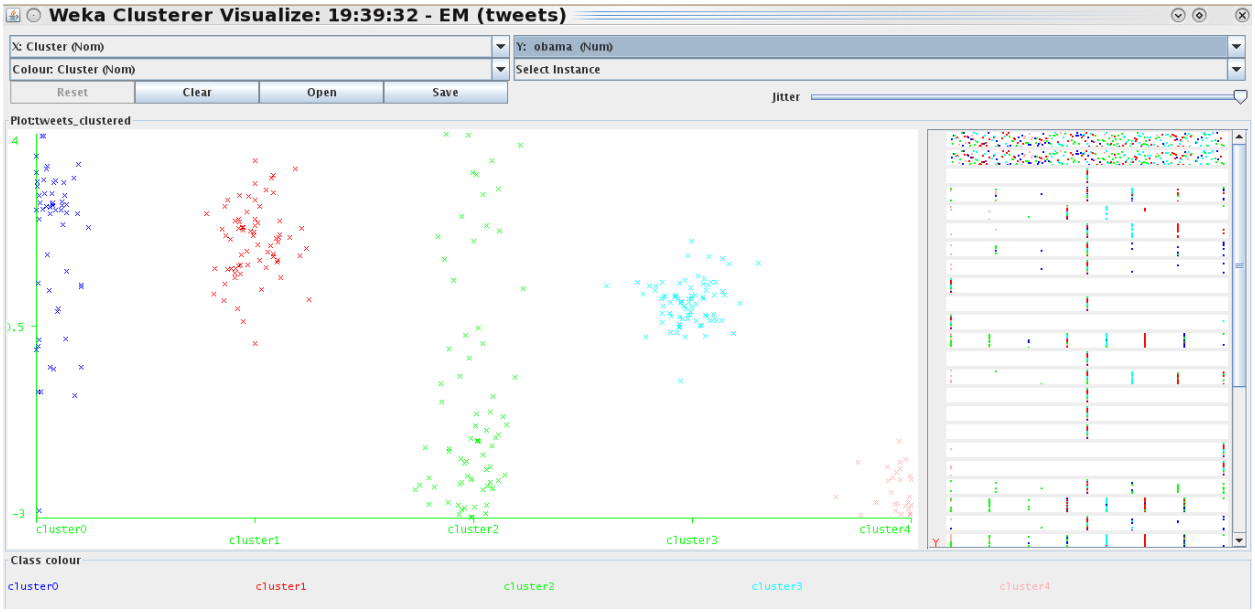

Figure 22 The distribution of clusters among the sentiment towards the topic “Obama”, where cluster 4 is the isolated cluster and 0 and 1 are another two isolated clusters on Weka’s visual.

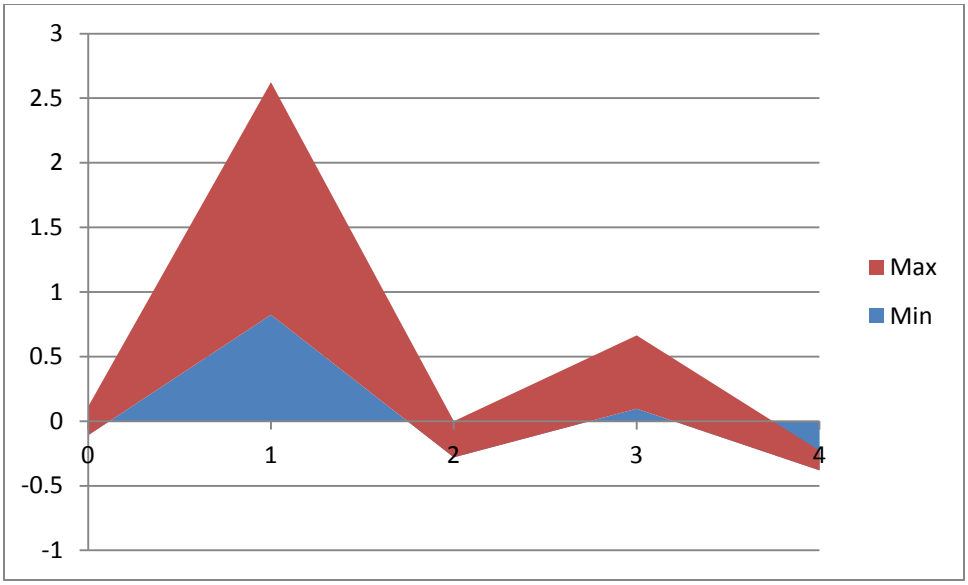

Figure 23 The distribution of clusters among the sentiment towards the topic “OWS”, where cluster 1 is the isolated cluster on an area plot using the minimum and maximum values.

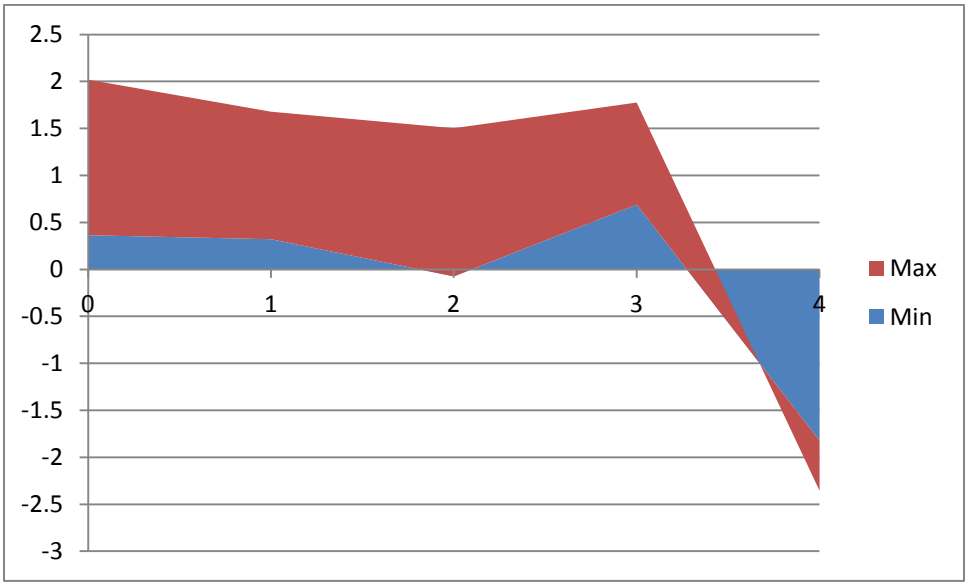

Figure 24 The distribution of clusters among the sentiment towards the topic “Romney”, where cluster 4 is the isolated cluster on an area plot using the minimum and maximum values.

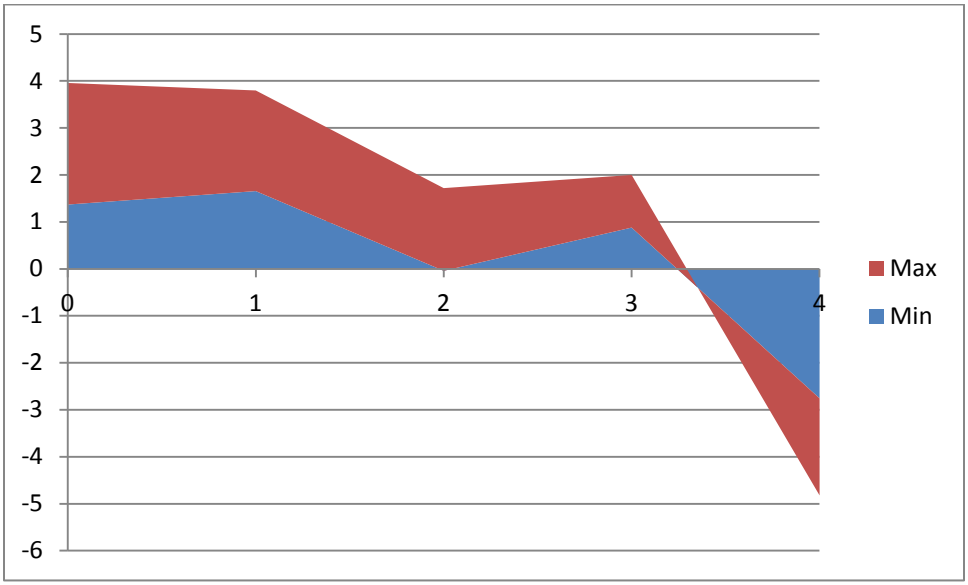

Figure 25 The distribution of clusters among the sentiment towards the topic “Obama”, where cluster 4 is the isolated cluster and 0 and 1 are another two isolated clusters on an area plot using the minimum and maximum values.

Table 8.3 shows the number of instances and significant percentages of referrers in isolated clusters corresponding to the particular topic and sentiment that caused the isolation. The total frequency of shown at the most right column is not the sum of all

channels' counts since some channels might be mentioned in the same tweet. Thus, we made a separate counter for counting the total.

Table 8.3 The number of mentions for each channel in topics with isolated clusters for experiment 2.

Topic>>Sentiment	ABC	NY times	Fox	CNN	Reuters	NBC
OWS > 0.82365						
General	0	0	1	21 (7%)	8	2
Cluster Specific(1)	0	0	1	6	8	1
Romney < -0.5339						
General	6	1	10	55 (53.3%)	12	7
Cluster Specific(4)	2	0	1	13	9	2
Obama < -2						
General	12	3	19 (6.3%)	105 (34.8%)	17 (5.6%)	19 (6.3%)
Cluster Specific(4)	5	0	2	29 (9.6%)	11	10
Obama > 1						
General	18 (5.9%)	6	27 (9%)	187 (62.1%)	31 (10.3%)	32 (10.6%)
Cluster Specific (0&1)	5	0	2	29 (9.6%)	11	10

Since CNN has the most significant percentages of influence assurance, we show the distribution graphs of referrers from all clusters among the sentiment towards "Occupy Wall Street", "Republicans" and "Obama", in figures 30-33. We can also notice from the figures the isolated clusters and the referrers' portions of them.

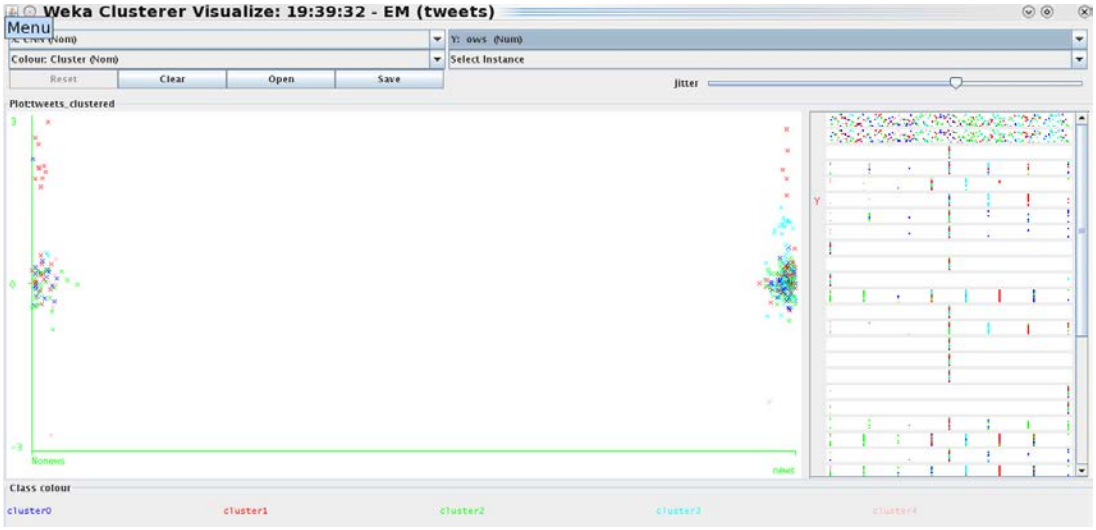

Figure 26 The distribution of clusters among the sentiment towards the topic “OWS” while showing the tweets which mentioned CNN, where cluster 1 is the isolated cluster on Weka’s visual.

Figure 27 The distribution of clusters among the sentiment towards the topic “Romney” while showing the tweets which mentioned CNN, where cluster 4 is the isolated cluster on Weka’s visual.

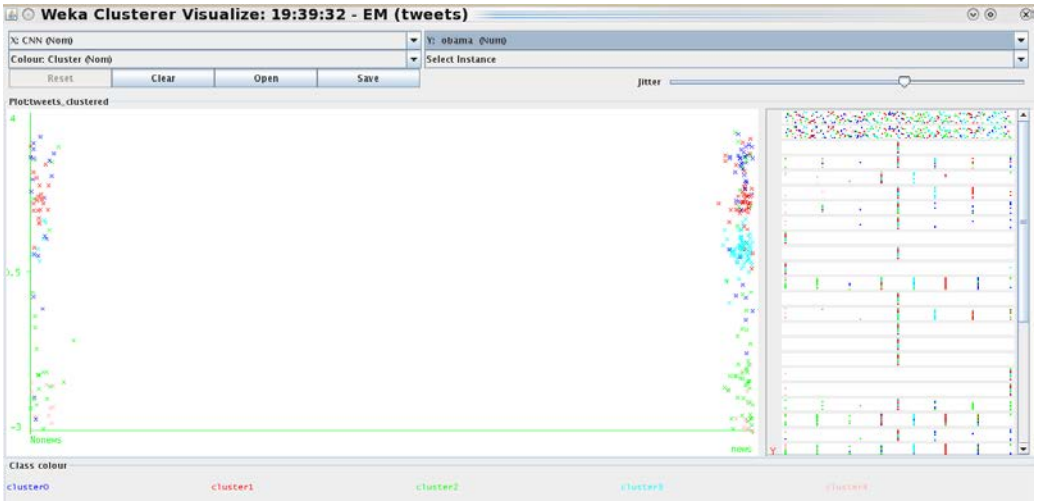

Figure 28 The distribution of clusters among the sentiment towards the topic “Obama” while showing the tweets which mentioned CNN, where cluster 4 is the isolated cluster on Weka’s visual.

EM uses discrete estimators for nominal attributes (just like naive Bayes does for classification). Weka's implementation of EM and naive Bayes assume that attributes are independent given the cluster/class. The numbers we see in the output for nominal attributes are frequency counts (Laplace corrected). Since EM is a soft clusterer (i.e. each instance belongs to each cluster probabilistically), the frequency counts can have fractional parts.

Experiment 3:

Here, we present the cluster distributions among the sentiment groups by probabilistic means. After detecting the isolated clusters, we calculate the percentage of news referrers out of these clusters in each topic to be compared with the percentages of referrers in isolated clusters used by the scoring method. This comparison is the validation process in which our inferences using the scoring sentiment are confirmed.

Using the same filters in experiment 1 but assigning the sentiment according to the sematic relatedness, here we apply the 3-topic filter, without restrictions for the news reference category. This filtering process only kept 1268 tweets with adjective group sentiment assignments.

This setting has resulted in 8 clusters selected, which took Weka 90.28 seconds. Table 9.1 shows the distribution of the instances among the clusters. Cluster number 3 is ignored by Weka, due to lack of instances contained by this cluster compared to other clusters, and distributed uniformly among the sentiment (i.e. cluster 3 has 6 instances each expressing different sentiment using the 6 groups for all topics).

Table 9.1 The percentages of distributions of clusters for experiment 3.

Cluster number	**Number of instances (percentage)**
0	133 (10%)
1	242 (19%)

2	82 (6%)
3	6 (0%)
4	230 (18%)
5	105 (8%)
6	26 (2%)
7	52 (4%)
8	398 (31%)

We present the distribution of the clusters among the sentiment groups used for each topic in table 9.2. For each topic, we mark the highest probabilistic values of each cluster with green, and then we red mark the values which do not share common sentiment with other clusters. This means we use the same notion of detecting isolated clusters to categorize them as segregated opinions.

Table 9.2 The distribution of clusters among the sentiment towards each topic using the mean and the standard deviation for experiment 3.

Sentiment Group (0-5)	**Cluster 0**	**Cluster 1**	**Cluster 2**	**Cluster 4**	**Cluster 5**	**Cluster 6**	**Cluster 7**	**Cluster 8**
election								
0	114.1018	170.6996	80.5723	209.1856	1.148	1.1298	1.1466	358.0164
1	1.0009	71.7353	4.228	1	1.0128	1.0007	1.0221	1.0001
2	23.2457	1.0013	1.0005	26.5778	1.0148	23.766	48.3938	1.0001
3	1.0007	1.0008	1.0003	1	107.2516	1.0001	1.0139	39.7326
4	1.0085	1.0045	1.0033	1.0001	1.2109	1.0014	1.771	1.0003
5	1.002	1.9914	1.0008	1.0001	1.044	1.0067	1.955	1.0001
[total]	141.3594	247.4329	88.8051	239.7637	112.6821	28.9048	55.3024	401.7495
vote								

0	47.8256	182.6382	43.5889	130.4153	74.8974	23.8979	48.4482	227.2885
1	1.0006	57.7889	41.2099	1	1.0002	1	1.0003	1.0001
2	89.5182	1	1.0003	105.348	1	1.0035	1.1299	1.0001
3	1.0005	1.0001	1.0003	1	33.538	1	1.0009	170.4602
4	1.0085	1.0045	1.0033	1.0001	1.2109	1.0014	1.771	1.0003
5	1.0061	4.0011	1.0025	1.0002	1.0356	1.0019	1.9521	1.0005
[total]	141.3594	247.4329	88.8051	239.7637	112.6821	28.9048	55.3024	401.7495
Romney								
0	135.3035	36.8575	79.2217	5.873	33.5586	1.0043	49.9049	152.2764
1	1.0002	178.5672	5.4317	1.0002	1.0003	1.0003	1.0001	1.0001
2	1.918	1.0003	1.0001	229.8229	1.0008	23.886	1.372	1.0001
3	1.0003	1.0003	1.0001	1.0004	75.1122	1.0001	1.0001	238.8864
4	1.1353	1.0337	1.1502	1.0653	1.0016	1.0061	1.022	7.5858
5	1.0021	28.9739	1.0013	1.0019	1.0086	1.0081	1.0034	1.0007
[total]	141.3594	247.4329	88.8051	239.7637	112.6821	28.9048	55.3024	401.7495
Obama								
0	2.0701	60.3493	1.618	5.9348	106.1791	21.0801	15.68	2.0887
1	1.0034	156.0513	81.9417	1.0002	1.0003	1	1.0026	1.0004
2	134.8733	1.0012	1.0043	229.7572	1.0002	3.8181	35.5424	1.0033
3	1.0057	1.0008	1.0029	1.0006	2.4997	1	1.0008	388.4895
4	1.4008	1.0434	2.2352	1.0689	1.0026	1.0062	1.0764	8.1665
5	1.0061	27.9869	1.003	1.002	1.0001	1.0004	1.0004	1.0012
[total]	141.3594	247.4329	88.8051	239.7637	112.6821	28.9048	55.3024	401.7495
media								
0	114.5908	228.6994	81.2645	210.7115	105.6949	23.8981	13.7661	366.3746
1	1.0021	14.7024	3.294	1	1	1	1.0011	1.0002
2	22.4721	1	1.0002	25.0223	1.0002	1.001	36.5041	1
3	1.0005	1.0003	1.0002	1	2.9499	1	1.0087	29.0404
4	1.292	1.0212	1.2454	1.0298	1.0016	1.0053	1.0706	3.3342
5	1.002	1.0095	1.0007	1	1.0355	1.0004	1.9518	1

[total]	141.3594	247.4329	88.8051	239.7637	112.6821	28.9048	55.3024	401.7495
republican								
0	131.0059	233.6567	9.659	220.3454	106.6877	23.9046	50.0323	343.7084
1	1.0002	9.7739	74.2258	1	1	1	1.0001	1
2	6.314	1	1.0011	15.4175	1	1.0001	1.2672	1
3	1.0002	1	1.0007	1	1.9942	1	1.0003	54.0046
4	1.0391	1.0023	1.9185	1.0007	1.0002	1	1.0027	1.0365
5	1	1	1	1	1	1	1	1
[total]	141.3594	247.4329	88.8051	239.7637	112.6821	28.9048	55.3024	401.7495

From this table we can observe that cluster number 5 expresses high concentration of using adjectives from group number 3, which is negatively biased group of adjectives, for the topic “elections”. Cluster number 0 expresses a positive sentiment using group 2 towards the “vote” topic. Using the same sentiment group, cluster number 6 expresses its positive opinion towards the “Media” topic. Cluster number 1 expresses also a positive sentiment but using sentiment group 1 towards the topic “Romney”. While cluster number 2 expresses its positive sentiment using sentiment group 1 towards the “Republican” topic. On the contrary, cluster number 7 expresses a negative sentiment towards the “Obama” topic using sentiment group 3.

We cannot compare those resulted clusters with the ones resulting from the scoring sentiment method, since the cross validation is totally different, which is indicated by the output number of clusters. The validation process could be only achieved when comparing the percentages of referrers in isolated clusters for each topic separately while considering the polarity of the sentiment concentrated by those isolated clusters.

According to the table, in this sense it is obvious in the figures 34-39 that topics “Elections”, “Vote”, “Media”, “Romney”, “Republicans” and “Obama” have different segregated unidimensional opinions. These figures show the clusters versus the sentiment towards each of the mentioned topics.

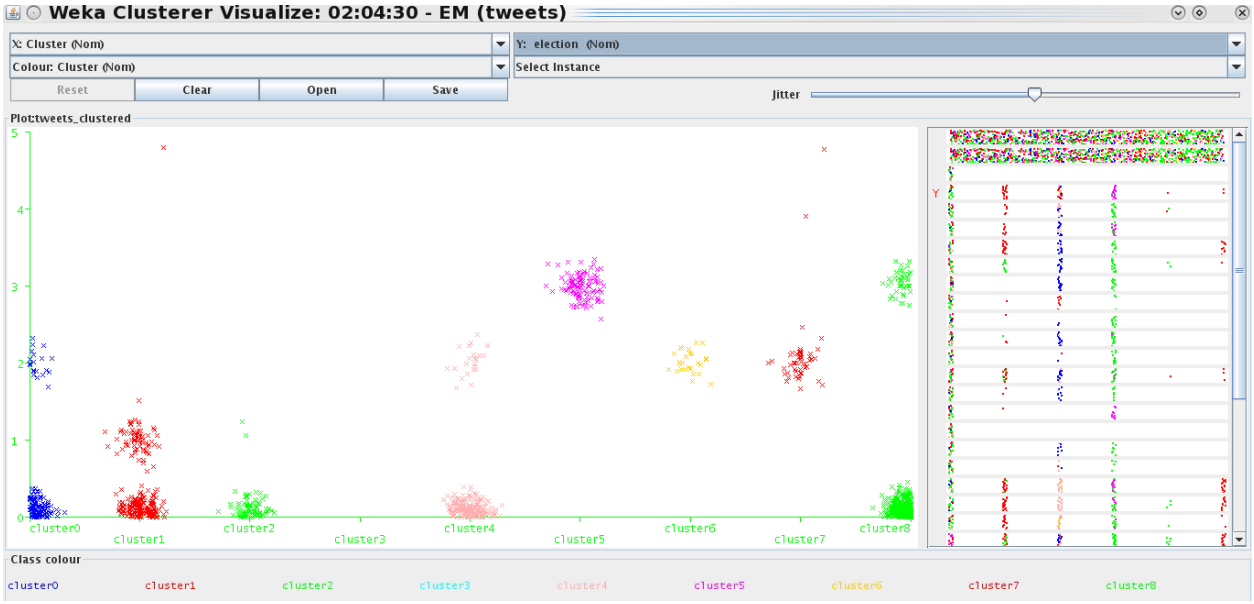

Figure 29 The distribution of clusters among the sentiment towards the topic “elections”, where cluster 5 is the isolated cluster and concentrated on sentiment group 3 on Weka’s visual.

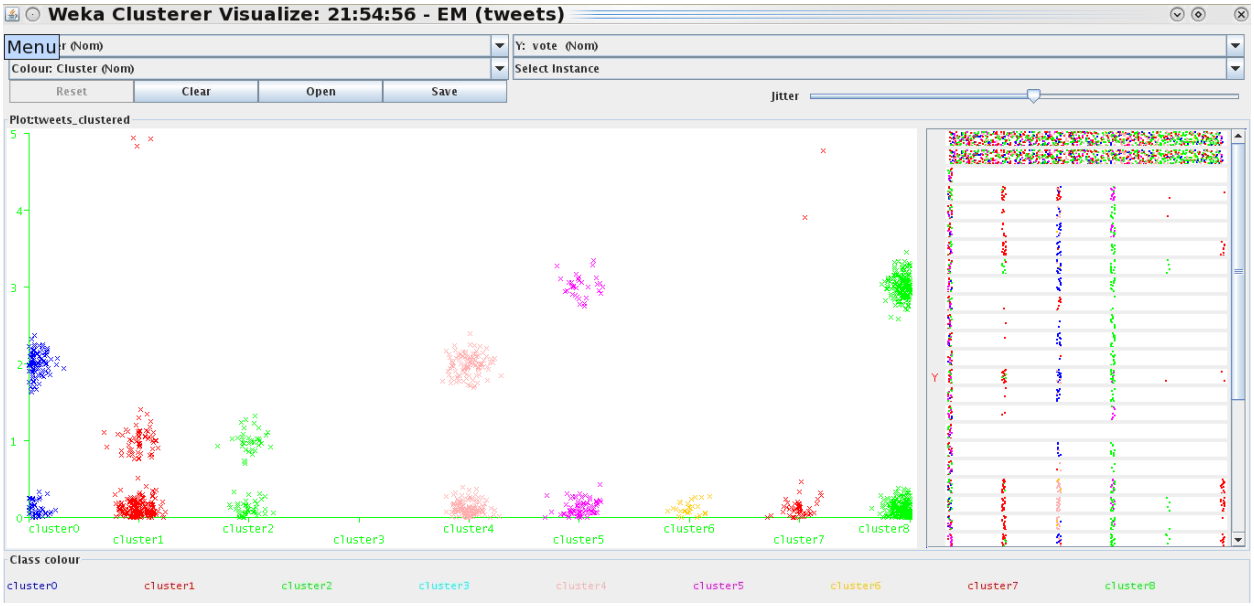

Figure 30 The distribution of clusters among the sentiment towards the topic “vote”, where cluster 0 is the isolated cluster and concentrated on sentiment group 2 on Weka’s visual.

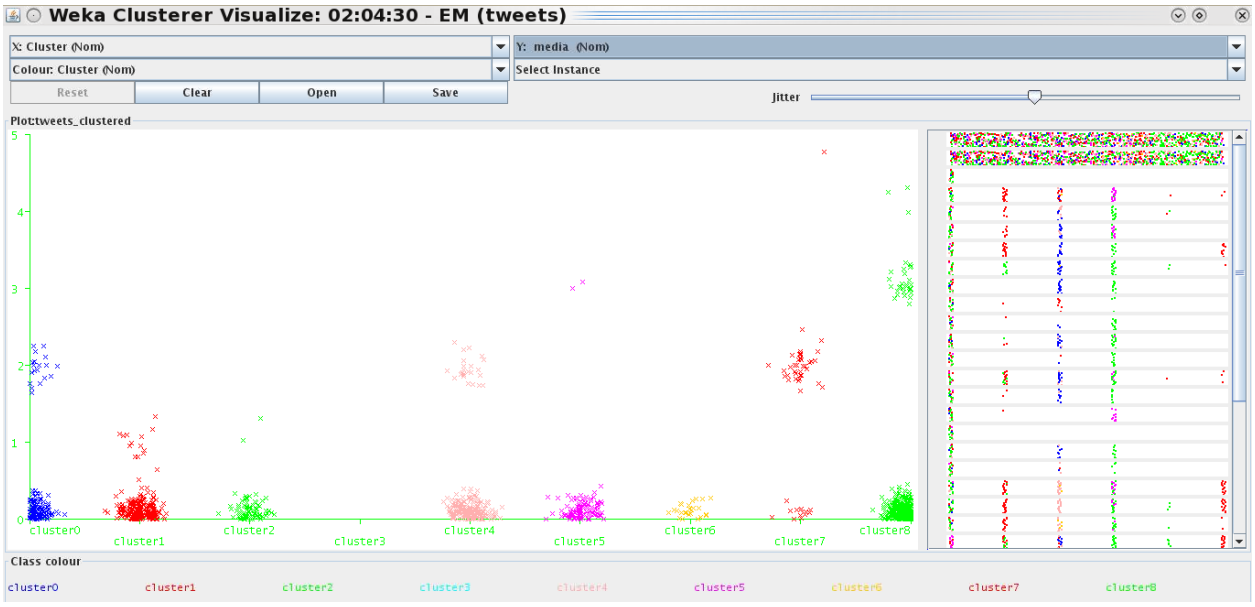

Figure 31 The distribution of clusters among the sentiment towards the topic “media”, where cluster 7 is the isolated cluster and concentrated on sentiment group 2 on Weka’s visual.

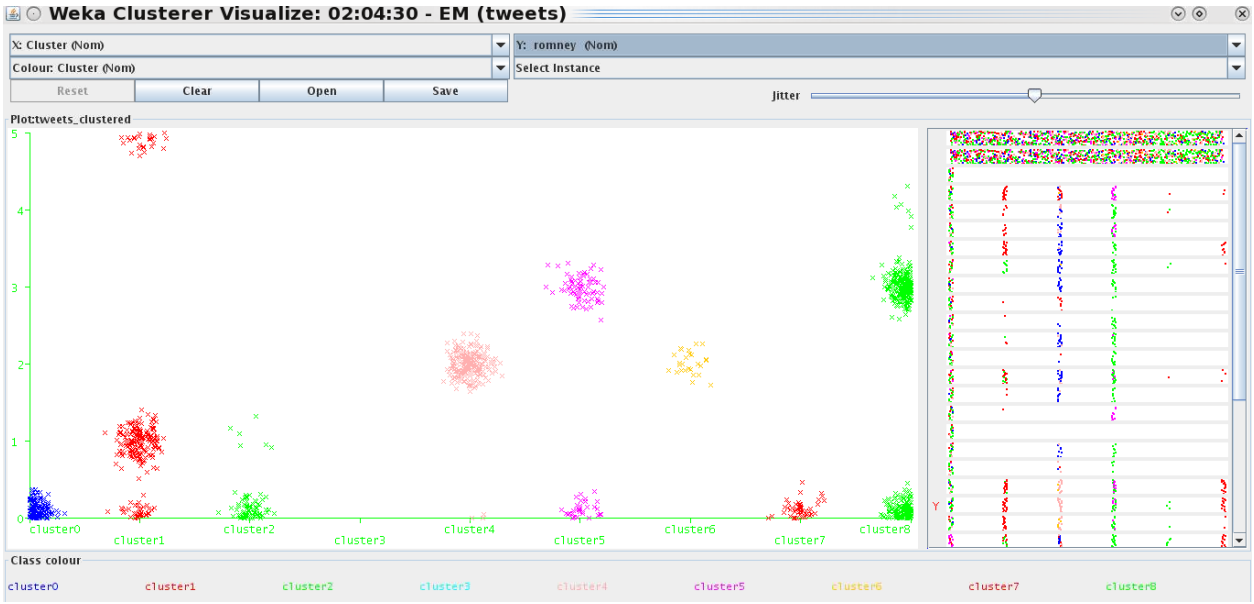

Figure 32 The distribution of clusters among the sentiment towards the topic “Romney”, where cluster 1 is the isolated cluster and concentrated on sentiment group 1 on Weka’s visual.

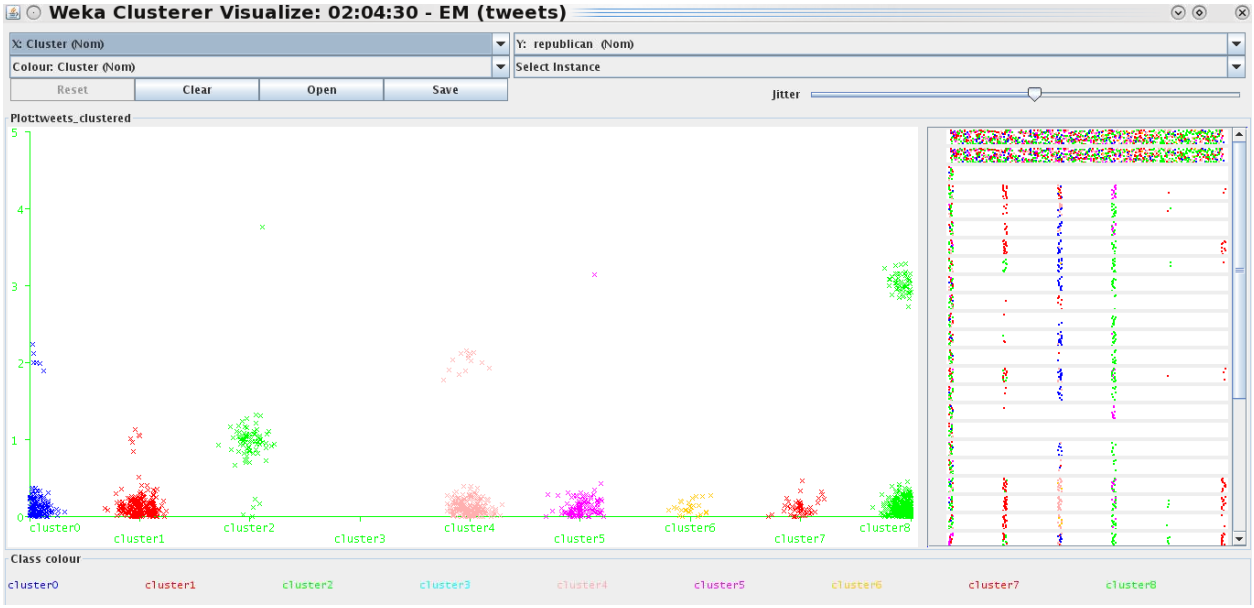

Figure 33 The distribution of clusters among the sentiment towards the topic "Republican", where cluster 2 is the isolated cluster and concentrated on sentiment group 1 on Weka's visual.

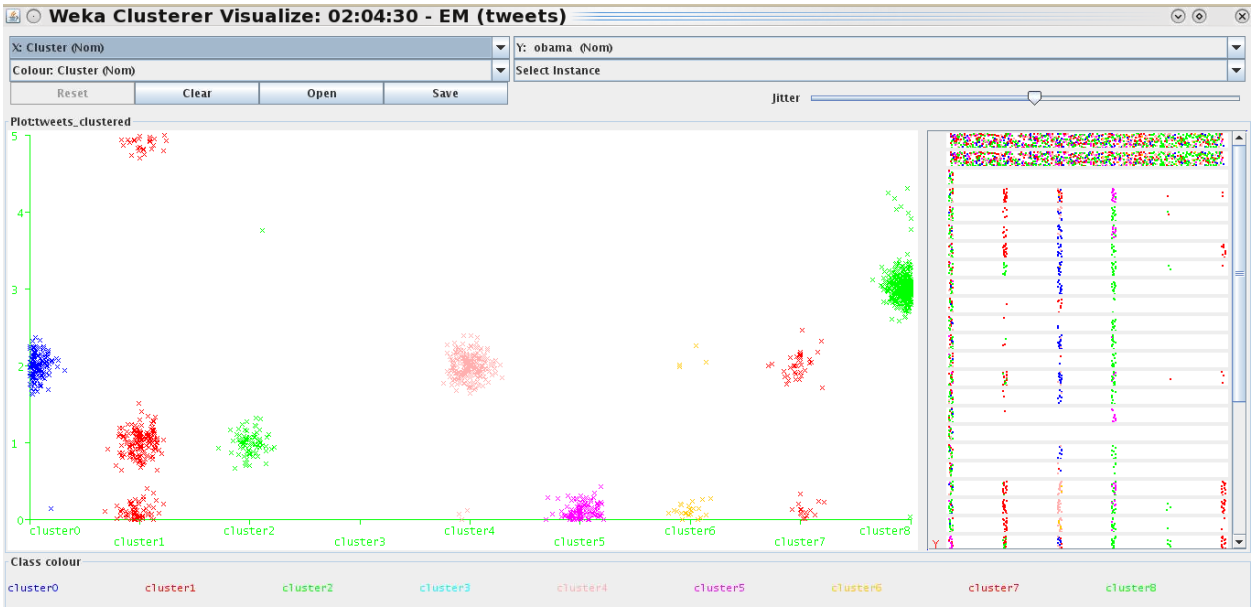

Figure 34 The distribution of clusters among the sentiment towards the topic “Obama”, where cluster 8 is the isolated cluster and concentrated on sentiment group 3 on Weka’s visual.

As per our definition to the influence, at the beginning of the chapter, we categories the influence into two types: general and cluster specific influence. The general influence assurance is the percentage of referrers of a channel from the total number of instances with the biased sentiment. While the cluster specific influence assurance is also the percentage of referrers of a channel from the total number of instances with the biased sentiment, but in the isolated cluster only. Table 9.3 shows the number of instances and the percentage of influence for each channel and all of them for both types of influences. We did not consider the topics that had general influence below 10%.

Table 9.3 The number of mentions for each channel in topics with isolated clusters for experiment 3.

Topic>>Sentiment	**ABC**	**NY times**	**Fox**	**CNN**	**Reuters**	**NBC**	**Total**
Media>>2							
General	2	1	0	4	0	0	7 (19.2%)
Cluster Specific	0	0	0	2	0	0	2 (5.5%)
Republicans>>1							
General	2	1	1	6	1	2	13 (17.6)
Cluster Specific	0	0	1	4	0	1	6 (8.1)
Obama>>3							

General	31	2	7	37	1	2	80 (20.6%)
Cluster Specific	29	1	7	35	0	1	73 (18.8%)

Figures 40-42 show the distribution of referrers and non-referrers of CNN of all clusters among the sentiment towards topics “Media”, “Republicans” and “Obama”. We show referrers of CNN, since they are the most influential.

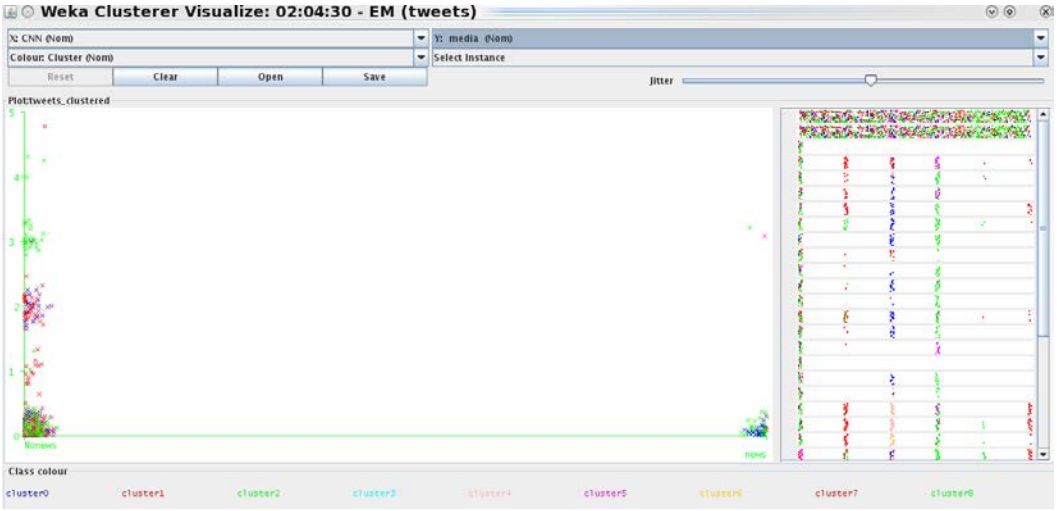

Figure 35 The distribution of clusters among the sentiment towards the topic “media” while showing the tweets which mentioned CNN, where cluster 7 is the isolated cluster on Weka’s visual.

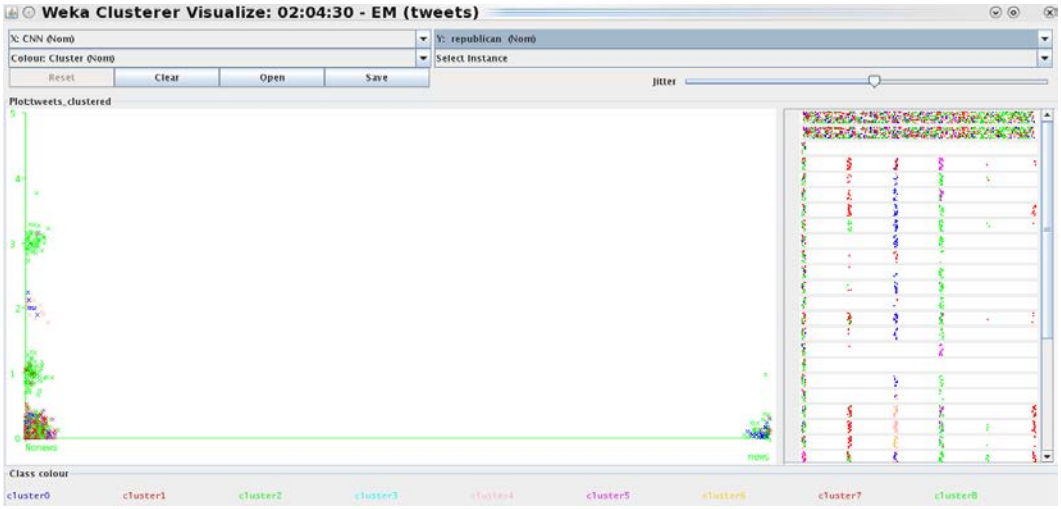

Figure 36 The distribution of clusters among the sentiment towards the topic "Republican" while showing the tweets which mentioned CNN, where cluster 2 is the isolated cluster on Weka's visual.

Figure 37 The distribution of clusters among the sentiment towards the topic “Obama” while showing the tweets which mentioned CNN, where cluster 8 is the isolated cluster on Weka’s visual.

Our observation from the figure 42 is that the isolated cluster 8 is concentrated at the sentiment group 3, while a big portion comparatively to the rest of the clusters is referring to CNN. This figure provides intuitive visual of the influence with the help of calculating the exact percentage to quantify the influence.

CHAPTER 5

CONCLUSION

Original contribution:

In summary, we proposed the challenge of measuring and quantifying the influence of mainstream media on Twitter users. The major assumptions for quantifying the measurement are based on the media social control theory, media bias theory and previous work done in defining the segregated opinions across the spectrum. The contribution towards this challenge is mainly about the framework and the model proposed. Basically, the framework proposed facilitated the basic input for our model, while the model is the main theme for detecting segregated opinions. The model depends totally on fitting the EM algorithm into finding the hidden variables, which are the sources of the opinions.

Methodology:

To test our framework and its model, we streamed-in tweets into our database on fedora and filtered the analyzed tweets according to three basic and two variable categories according to each experiment setting. We defined the trending topics as the frequent itemsets that are the output from the Apriori algorithm. The sentiment values where assigned using scores and semantic relatedness between adjectives used. The semantic relatedness is described through the hierarchical structure of adjectives, when the

hierarchical clustering algorithm is applied on the lexicon dissimilarity matrix of the adjectives. The sentiment matrix is the output from the sentiment assignment step and the input for the opinion clustering step. The EM algorithm is applied on the sentiment matrix as the observed variables to find the hidden variables' parameters, the cluster parameters, which are the sources of the opinions. In order to characterize the anonymous sources of opinions we calculate the percentages of news mentions within all and the isolated clusters. We only consider the news mentions within the tweets which showed sentiment below the minimum or above the maximum of the isolated clusters' ranges.

Main findings:

In our three experiments, we used different setups of filtering categories, where two of them are similar in the used categories but different in the sentiment assignment. First, we filtered out the RTs to analyze original messages only, and the tweets which have no adjectives and/or less than three topics. The output result from this setup is 10 clusters which is the maximum number Weka could reach, since the training set is split randomly into 10 folds. The alternating EM process is applied 10 times maximum to increase the clusters by 1 incrementally each step starting from 1 cluster. However, the resulting isolated clusters showed insignificant percentage of tweets mentioning news channels. Thus, we change the 3 topic filter to be 1 and added the news filter, in order to focus on the tweets which mentioned the news channels only. For this setup, the isolated clusters

showed significant percentage of tweets mentioning the news channels. Lastly, we repeated the first setup but by assigning the sentiment using the semantic relatedness of adjectives. The isolated clusters also showed significant percentage in news mentions.

Future work:

We plan to use the association rules between the news channels and the most frequent 30 words in searching the web news archives for articles. By these keywords we optimize the finding of more articles related to twitter user's interests. We would apply the same sentiment analysis techniques on tweets and visualize them in comparison to the current results as a validation step. Additionally, a better idea is to visualize the sentiment versus time of these articles in comparison with the tweets, since we have the tweets' timestamps.

Disadvantages:

The disadvantage in our framework is the filtering of tweets which contain more than one adjective, since we were not able to differentiate the reference of adjectives to different nouns (topics) within the same tweet. However, in the future work we plan to use the NLTK to understand how can we differentiate between more than one adjective references using the sentence structure.

REFERENCES

A. Lourenço, M. Conover, A. Wong, F. Pan, Alaa Abi-Haidar, A. Nematzadeh, H. Shatkay, and L.M. Rocha. “Testing Extensive Use of NER tools in Article Classification and a Statistical Approach for Method Interaction Extraction in the Protein-Protein Interaction Literature” Proceedings of the BioCreative III Workshop 2010, Bethesda, Maryland, September 13-15, 2010.

Ah-hwee Tan, “Text Mining: The state of the art and the challenges”, In Proceedings of the PAKDD Workshop on Knowledge Discovery from Advanced Databases, pp. 65-70,1999

Aktolga, Elif, and James Allan. “Sentiment diversification with different biases.” Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. ACM, 2013

Alisa Kongthon, “A Text Mining Framework for Discovering Technological Intelligence to Support Science and Technology Management”, Technical Report, Georgia Institute of Technology, April 2004

Béatrice Daille and Eric Gaussier and Jean-Marc Langé. “Towards Automatic Extraction of Monolingual and Bilingual Terminology”. 1994.

Becker, Hila, Mor Naaman, and Luis Gravano. “Beyond Trending Topics: Real-World Event Identification on Twitter.” ICWSM 11 (2011): 438-441

Bench-Capon, Frans, Coenen, and Leng, “An Experiment in Discovering Association Rules in the Legal Domain”, In Proceedings of the Workshop on Legal Information Systems and Applications, pp. 1056–1060, 2000

Bollen, Johan, Huina Mao, and Xiaojun Zeng. “Twitter mood predicts the stock market.” Journal of Computational Science 2.1 (2011): 1-8

Cappelli, Carmela. “Identifying word senses from synonyms: a cluster analysis approach.” Quaderni di Statistica 5 (2003): 105-117

Ceppellini, R., Siniscalco, M. & Smith, C.A. Ann. Hum. Genet. 20, 97–115. Article, PubMed, ISI, ChemPort. 1955

Chen, Zhiyuan, et al. “Discovering coherent topics using general knowledge.” Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. ACM, 2013

Chen, Zhiyuan, et al. “Identifying Intention Posts in Discussion Forums.” Proceedings of NAACL-HLT. 2013

Chengqi Zhang, Shichao Zhang. “Association Rule Mining: Models and Algorithms” ISBN-13: 9783540435334, Springer Berlin Heidelberg, 5/28/2002.

Chenn-Jung,Huang,Jia-Jian,Liao,Dian-Xiu,Yang,Tun-Yu,Chang,Yun-Cheng Luo, "Realization of a news dissemination agent based on weighted association rules and text mining techniques", Journal Expert Systems with Applications,Vol.37,no.9, September, 2010

D. Janetzko, H. Cherfi, R. Kennke, A. Napoli, and Y. Toussaint. Knowledge-based selection of association rules for text mining. In Proceedings of ECAI'2004, 2004

D.D. Lewis, "An evaluation of phrasal and clustered representations on a text categorization task", In proceedings of SIGIR, pp: 37-50, 1992

D'Alessio & Allen "Selective exposure and dissonance after decisions." 2002.

Delgado, Martin-Bautista, Sanchez and Vila, "Mining text data: special features and patterns", In proceedings of EPS Exploratory workshop on pattern Detection and Discovery in Data mining, London, UK, September 2002

DeMarzo, Peter M., Dimitri Vayanos, and Jeffrey Zwiebel. "Persuasion bias, social influence, and unidimensional opinions." The Quarterly Journal of Economics 118.3 (2003): 909-968.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incom plete data via the EM algorithm. Journal of the Royal Statistical Society, Series B, 39(1):1-38.

Edda and Jorg, “Text categorization with support vector machines. How represent texts in input space?”, Machine Learning, Vol.46, pp.423-444, 2002

Eric Brill. “A Simple Rule-Based Part of Speech Tagger”. 1992.

Fatudimu, Musa, Ayo and Sofoluwe, “Knowledge Discovery in Online Repositories: A Text Mining Approach”, European Journal of Scientific Research, Vol.22 No.2, pp.241-250, 2008.

Fei Wu, Daniel S. Weld, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics,pp.118-127,2010

Fei, Geli, et al. “Exploiting Burstiness in Reviews for Review Spammer Detection.” Seventh International AAAI Conference on Weblogs and Social Media. 2013

Feldman, Fresko, Kinar, Lindell, Liphstat, Rajman, Schler and Zamir, “Text mining at the term level”, In proceedings of 2nd European Symposium of Principles of Data mining and Knowledge Discovery, pp.65-73, 1998

Festinger, Leon “A Theory of Cognitive Dissonance. California: Stanford University Press.” 1957.

H. Lu, L. Feng, and J. Han. Beyond intratransaction association analysis mining multidimensional intertransaction association rules. ACM Trans. Inf. Syst., 18(4):423-454, 2000

Hall, Stuart. 1973. Encoding and decoding in the television discourse. Birmingham, England: Centre for Cultural Studies, University of Birmingham

Hany Mahgoub, “Mining Association rules from unstructured documents”, World Academy of Science, Engineering and Technology, Vol.20, No.1, pp.1-6, 2006

Hany Mahgoub, Dietmar Rösner, Nabil Ismail and Fawzy Torkey, “A Text Mining Technique Using Association Rules Extraction”, International Journal Of Computational Intelligence, Vol.4, No.1, pp.21- 28, 2008

Herman, Edward S. and Noam Chomsky Manufacturing Consent: The Political Economy of the Mass Media. New York: Pantheon. 1998.

Hisham Al-Mubaid and Rajit K Singh, “A New Text Mining Approach for Finding Proteinto-Disease Associations”, American Journal of Biochemistry and Biotechnology, Vol.1, No.3, pp.145-152, 2005

Hotho, Nurnberger and Paass, “A Brief Survey of Text Mining Export”, LDV Forum, Vol.20, No.2, pp.19-62, 2005

J. F. Roddick and M. Spiliopoulou. Survey of temporal knowledge discovery paradigms and methods. IEEE Transactions on Knowledge and Data Engineering, 14(4):750-767, 2002

K. H. Tung, H. Lu, J. Han, and L. Feng. Efficient mining of intertransaction association rules. IEEE Transactions on Knowledge and Data Engineering, 15(1):43-56, 2003

Kim, Seungyeon, et al. “Beyond sentiment: The manifold of human emotions.” arXiv preprint arXiv:1202.1568 (2012)

Kodratoff, “Knowledge discovery in texts: A Definition and Applications”, Proc. of the 11th International Symposium on Foundations of Intelligent Systems, pp.16-29, 1999

Kwak, Haewoon and Lee, Changhyun and Park, Hosung and Moon, Sue “What is Twitter, a social network or a news media?” WWW '10: Proceedings of the 19th international conference on World wide web, ACM, 591—600. New York, NY, USA. 2010.

Landeau, Aumann, Feldman, Fresko, Lindell, Lipshat and Zamir, “TextVis: An integrated Visual Environment for Text mining”, In Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'98), Nantes, September 1998

Latiri, Cherif Chiraz, and Sadok Ben Yahia. “Textmining: Generating association rules from textual data.” INFORSID. 2001.

Lebart L., Salem A., Berry L. Exploring Textual Data, Kluwer Academic Publishers, Dordrecht. 1998.

Liang-Chih Yu, Chien-Lung Chan, Chao-Cheng Lin, I-Chun Lin, “Mining association language patterns using a distributional semantic model for negative life event classification”, Journal of Biomedical Informatics,Vol.44,no. 4, August, 2011

Linhao Zhang. “Sentiment Analysis on Twitter with Stock Price and Significant Keyword Correlation”. 2013

Lumezanu, Cristian, Nick Feamster, and Hans Klein. “# bias: Measuring the Tweeting Behavior of Propagandists.” Sixth International AAAI Conference on Weblogs and Social Media. 2012

M. Dunham. Data Mining: Introductory and Advanced Topics. Prentice Hall, 2003

Manning and Schütze, “Foundations of statistical natural language processing”, MIT Press 1999

Martin Rajman and Romaric Besançon. “Text Mining - Knowledge extraction from unstructured textual data”. 6th Conference of International Federation of Classification Societies (IFCS-98), 473--480. 1998.

Michelle de Haaff (2010), Sentiment Analysis, Hard But Worth It!, Customer Think, retrieved 2010-03-12

Mosley Jr, Roosevelt C. “Social media analytics: Data mining applied to insurance Twitter posts.” Casualty Actuarial Society E-Forum, Winter 2012 Volume 2. 2012

Mukherjee, Arjun, and Bing Liu. “Mining contentions from discussions and debates.” Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012

Myers, Seth A., Chenguang Zhu, and Jure Leskovec. “Information diffusion and external influence in networks.” Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012.

Naaman, Mor, Hila Becker, and Luis Gravano. “Hip and trendy: Characterizing emerging trends on Twitter.” Journal of the American Society for Information Science and Technology 62.5 (2011): 902-918

Nasukawa and Nagano, “Text Analysis and Knowledge Mining System”, IBM Systems Journal, Vol.40, No.4, pp.967-984, October 2001

New Internationalist Magazine. June, 1981.The Big Four Retrieved 02 Mar 2013

P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In Proceedings of KDD’2002, 2002

Pablo F. Matos , Leonardo O. Lombardi, Thiago A. S. Pardo, Cristina D. A. Ciferri, Marina T. P. Vieira, and Ricardo R. Ciferri, “An Environment for Data Analysis in Biomedical Domain: Information Extraction for Decision Support Systems,” Proceedings of the 23rd international conference on Industrial engineering and other applications of applied intelligent systems - Volume Part I,pp.306-316,2010

Pak Chung Wong, Paul Whitney and Jim Thomas, “Visualizing Association Rules for Text Mining”, in proceedings of IEEE symposium on Information Visualization”, pp.120, 1999

Qiankun Zhao, Sourav S. Bhowmick “Association Rule Mining: A Survey”, Technical Report, CAIS, Nanyang Technological University, Singapore, No. 2003116, 2003

R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules,” Jorge B. Bocca, Matthias Jarke, Carlo Zaniolo, eds., in proceedings of 20th International Conference on Very Large Data Bases, pp. 487-499, Santiago, Chile, 1994

Rajman, Martin, and Romaric Besançon. “Text mining-knowledge extraction from unstructured textual data.” Advances in Data Science and Classification. Springer Berlin Heidelberg, 1998. 473-480

Raymond J. Mooney and Razvan Bunescu, “Mining knowledge from text using information extraction”, ACM SIGKDD Explorations Newsletter, Vol.7,No.1, pp.3-10, 2005

Ricardo Baeza-Yates, Alistair Moffat and Gonzalo Navarro, “Searching large text collections”, Handbook of massive data sets, pp.195-243, ISBN: 1-4020-0489-3, 2002

Rodrigues Barbosa, Glívia Angélica, et al. “Characterizing the effectiveness of twitter hashtags to detect and track online population sentiment.” Proceedings of the 2012 ACM

annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts. ACM, 2012

SAS Institute “The CLUSTER Procedure: Clustering Methods”. SAS/STAT 9.2 Users Guide. Retrieved 2009-04-26.

Sebastiani, “Machine Learning in automated text categorization”, ACM Computing Surveys, Vol.34, No.1, pp.1-47, 2002

Shatkay and Feldman, “Mining the Biomedical Literature in the Genomic Era: An Overview”, Journal of Computational Biology, Vol.10, No.6, pp.821-855, 2003

Sheng-Tang Wu, Yuefeng Li and Yue Xu, “Deploying Approaches for Pattern Refinement in Text Mining”, in proceedings of the Sixth IEEE International Conference on Data Mining, pp. 1157-1161, 2006

Sophia Ananiadou, Sampo Pyysalo, Jun’ichi Tsujii, Douglas B. Kel,”Event extraction for systems biology by text mining the literature,”Trends in Biotechnology,vol.28,no.7,pp.381-390,2010

Suneetha Manne,Dr. S. sameen Fatima, “A Novel Approach for Text Categorization of Unorganized data based with Information Extraction” International Journal on Computer Science and Engineering (IJCSE),Vol. 3 No. 7,July 2011

Valentina Ceausu and Sylvie Despres, “Text Mining Supported Terminology Construction”, In proceedings of the 5th International Conference on Knowledge Management, Graz, Austria, 2005

VDS Baghela, Dr. S.P. Tripathi, “A Survey on Association Rules in Case of Multimedia Data Mining,” International Journal of Computer Science and Technology,vol.3, Issue 1 , pp.649-652, March 2012

Vishal Gupta and Gurpreet S. Lehal, “A Survey of Text Mining Techniques and Applications”, Journal Of Emerging Technologies in Web Intelligence, Vol.1, No.1, pp.60-76, August 2009

Wang, Chi, et al. “Constructing Topical Hierarchies in Heterogeneous Information Networks.”

Wang, Jing, et al. “Diversionary comments under political blog posts.” Proceedings of the 21st ACM international conference on Information and knowledge management. ACM, 2012

Wei, Zhongyu, et al. “Mainstream media behavior analysis on Twitter: a case study on UK general election.” Proceedings of the 24th ACM Conference on Hypertext and Social Media. ACM, 2013.

Wiebe, Janyce, et al. “Recognizing and Organizing Opinions Expressed in the World Press.” New Directions in Question Answering. 2003

Yang, Xintian, et al. “A framework for summarizing and analyzing twitter feeds.” Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012

Yiye Ruan, Hemant Purohit, David Fuhry, Srinivasan Parthasarathy, Amit Sheth “Prediction of Topic Volume on Twitter” In WebSci 2012

Yue Dai, Tuomo Kakkonen, Erkki Sutinen, “MinEDec: a Decision-Support Model That Combines Text- Mining Technologies with Two Competitive Intelligence Analysis Methods,” International Journal of Computer Information Systems and Industrial Management Applications,vol.3pp.165-173,2011

Zhang, Lei, and Bing Liu. “Aspect and Entity Extraction for Opinion Mining.” Data Mining and Knowledge Discovery for Big Data. Springer Berlin Heidelberg, 2014. 1-40.

Lars Kai Hansen, Adam Arvidsson, Finn Nielsen, Elanor Colleoni, Michael Etter, "Good Friends, Bad News - Affect and Virality in Twitter", The 2011 International Workshop on Social Computing, Network, and Services. SocialComNet 2011.